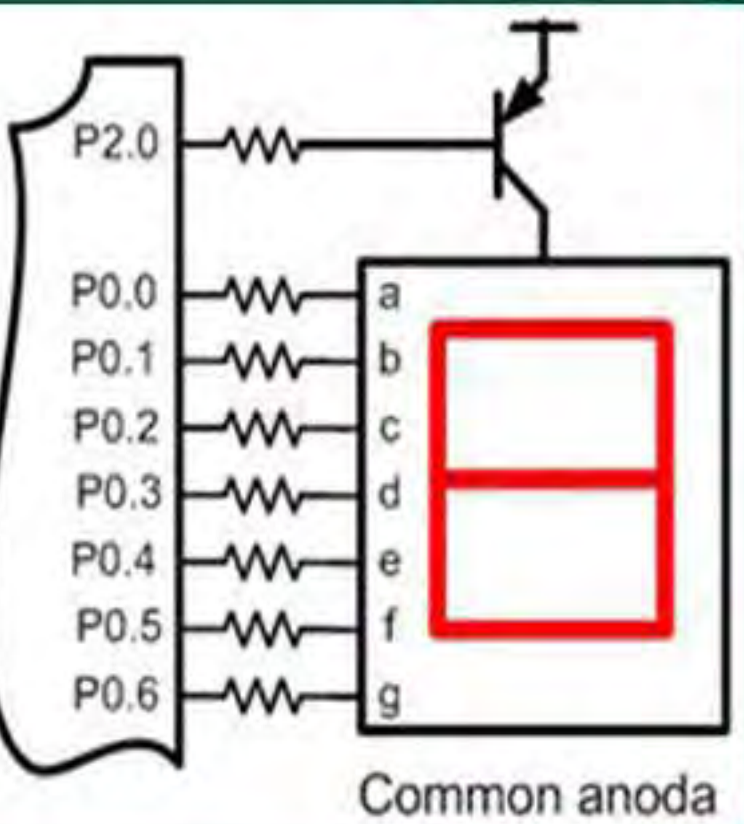
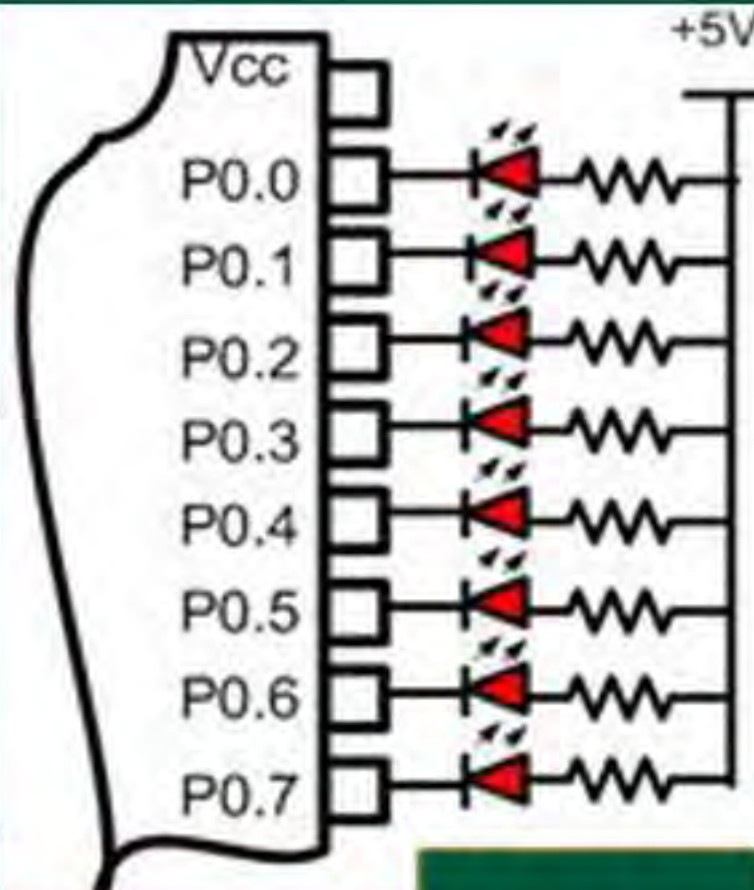
KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
REPUBLIK INDONESIA
2013

Sistem Kontrol Terprogram





Semester 5

Kelas XI

PENULIS

KATA PENGANTAR

Puji syukur kehadirat Alloh SWT, dengan tersusunnya buku Teknik Sistem Kontrol Terprogram untuk kelas XII semester 1 ini, semoga dapat menambah khasanah referensi khususnya di bidang tekologi industri yang akhir-akhir ini berkembang begitu pesatnya di Indonesia.

Isi buku ini sengaja disajikan secara praktis dan lengkap sehingga dapat membantu para siswa Sekolah Menengah Kejuruan (SMK), guru serta para praktisi industri. Teknik Sistem Kontrol Terprogram untuk kelas XII semester 1 yang selama ini dideskripsikan secara variatif dan adaptif terhadap perkembangan serta kebutuhan berbagai kalangan praktisi industri. Penekanan dan cakupan bidang yang dibahas dalam buku ini sangat membantu dan berperan sebagai sumbangsih pemikiran dalam mendukung pemecahan permasalahan yang selalu muncul didalam mempelajari dasar-dasar instrumentasi.

Oleh karena itu, buku ini disusun secara integratif antar disiplin ilmu yaitu Teknik Sistem Kontrol Terprogram untuk kelas XII yang saling mendukung sehingga skill yang diperlukan terkait satu dengan lainnya.

Tim penulis mengucapkan terima kasih kepada berbagai pihak yang telah membantu materi naskah serta dorongan semangat dalam penyelesaian buku ini. Kami sangat berharap dan terbuka untuk masukan serta kritik konstruktif dari para pembaca sehingga dimasa datang buku ini lebih sempurna dan implementatif.

DAFTAR ISI

Contents

DAFTAR GAMBAR

BAB 1

KEGIATAN BELAJAR 1 :
SISTEM KONTROL

A. Tujuan Pembelajaran

Setelah menyelesaikan kegiatan belajar 1, siswa diharapkan mampu :

1. Mampu mendefinisikan dan memahami pengertian kontrol.
2. Mampu mendefinisikan dan memahami klasifikasi sistem kontrol.
3. Mampu mendefinisikan dan memahami sistem kontrol otomatis.
4. Mampu mendefinisikan dan memahami macam-macam sistem kontrol.

B. Uraian Materi

Sesuai dengan namanya, sistem kontrol atau biasa juga disebut dengan sistem pengaturan adalah sebuah sistem dimana beberapa besaran fisik diatur, diubah dan dimanipulasi dengan mengatur besaran input (masukan). Sebuah sistem didefinisikan sebagai sekumpulan perangkat yang dirakit untuk membentuk sebuah

perangkat gabungan yang dapat menghasilkan sebuah fungsi yang spesifik.

Sebuah sistem kontrol yang ideal adalah sebuah sistem dimana besaran output (keluaran) merupakan fungsi langsung dari besaran input (masukan).



Gambar 1 Blok diagram sistem kontrol ideal.

1. Pengertian Sistem Kontrol

Sistim Kontrol didefinisikan sebagai sekumpulan perangkat yang dirakit untuk membentuk sebuah perangkat gabungan yang dapat menghasilkan sebuah fungsi keluaran spesifik yang diinginkan untuk mengatur sebuah besaran tertentu.

2. Klasifikasi Sistem Kontrol

Berdasarkan Berdasarkan definisi umum tersebut, terdapat beberapa metoda untuk mengklasifikasikan sebuah sistem kontrol yaitu berdasarkan :

a. Mode pengaturan plant
   1) Mode pengaturan ON-OFF (misalnya : kontrol relay, PLC)
   2) Mode pengaturan kontinyu (misalnya : PID *Control*)

b. Penggunaan teknik umpan balik
   1) Sistem kontrol loop terbuka
   2) Sistem kontrol loop tertutup

c. Teknik Pengolahan Data

1) Teknik pengolahan data analog

2) Teknik pengolahan data digital

d. Aplikasi

1) Sistem kontrol sekuensial

2) Sistem kontrol numerik

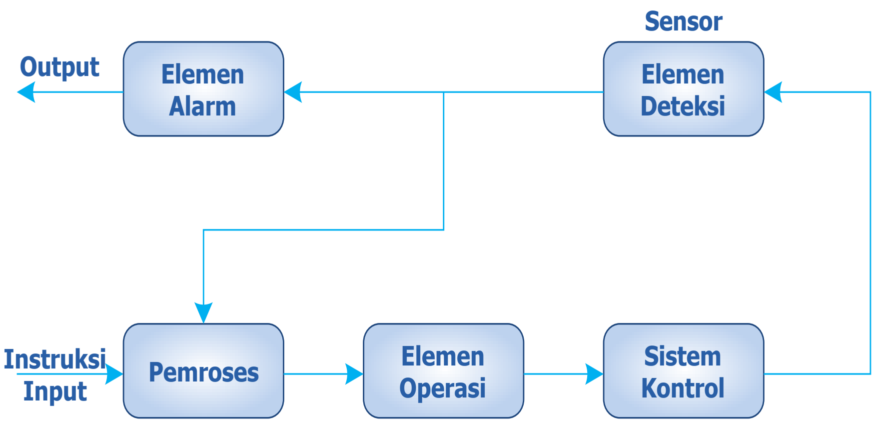
3) Sistem kontrol proses

4) Servomekanis (*servomechanism*)

3. Sistem kontrol otomatis (*Automatic Control*)

a. Sekuensi Control

Contoh Sekuensi diproses berdasarkan langkah-langkah (step-step, satu persatu) sesuai dengan kebutuhan.

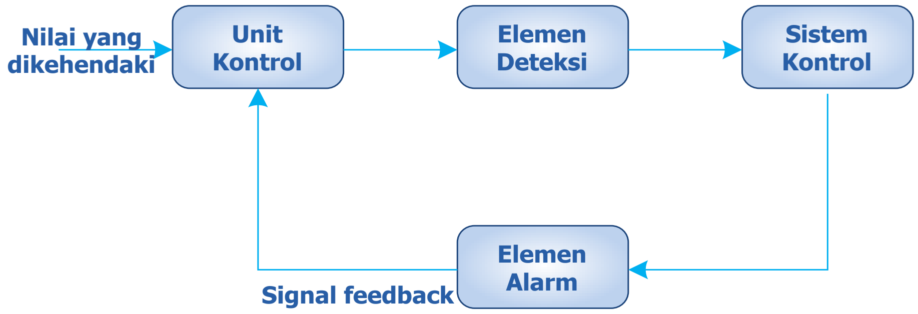


Gambar 2 Blok diagram komposisi sekuensial kontrol.

b. Feedback Control

Feedback Control adalah dimana signal output diumpankan kambali ke input, dengan membandingkan hasil pengukuran

yang dikehendaki untuk melakukan koreksi sehingga seimbang keduanya.



Gambar 3 Blok diagram sekuensial *feedback control*

4. Macam-macam sekuensial kontrol

a. Relay sekuensial kontrol

b. Logik sekuensial kontrol

c. *Programmable control* rangkaian mikro komputer

5. Tipe-tipe kontrol

a. Kontrol dengan relay

Kemampuan :

1) Kekuatan singkat

2) Keandalan rendah

3) Fungsi kontrol terbatas

b. Kontrol tanpa relay

Umumnya mengunakan komponen semi konduktor

c. Kontrol dengan IC digital

d. Programable Kontroller

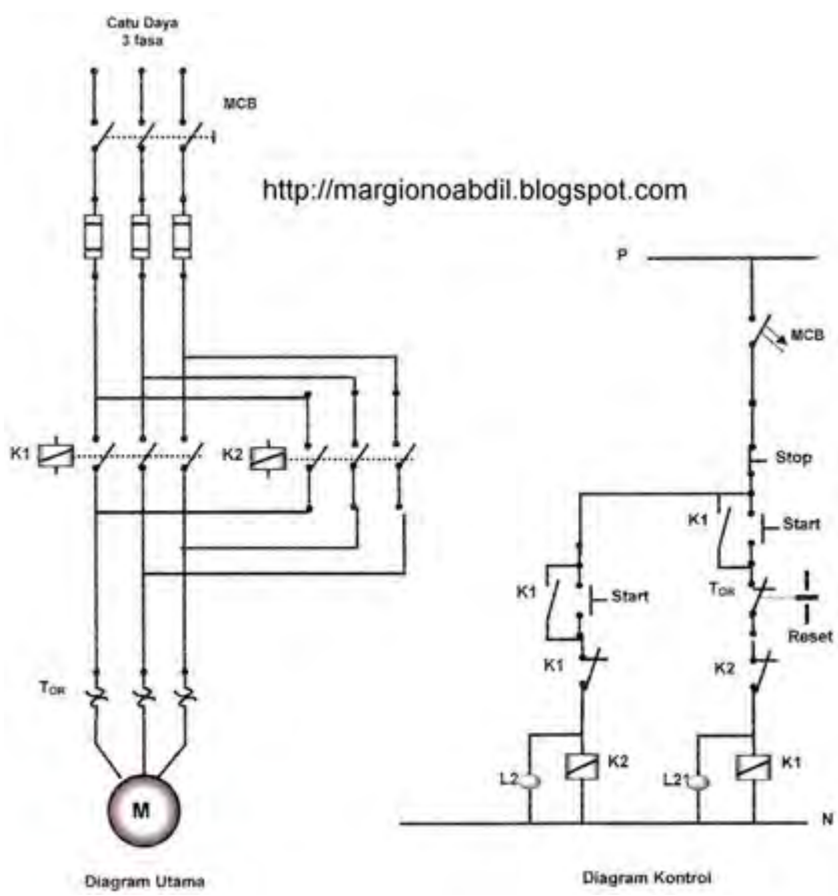
Suatu tipe komputer yang dirancang khusus untuk kontrol sekuensial (kontrol dengan program).

e. Kontrol melalui mikro computer

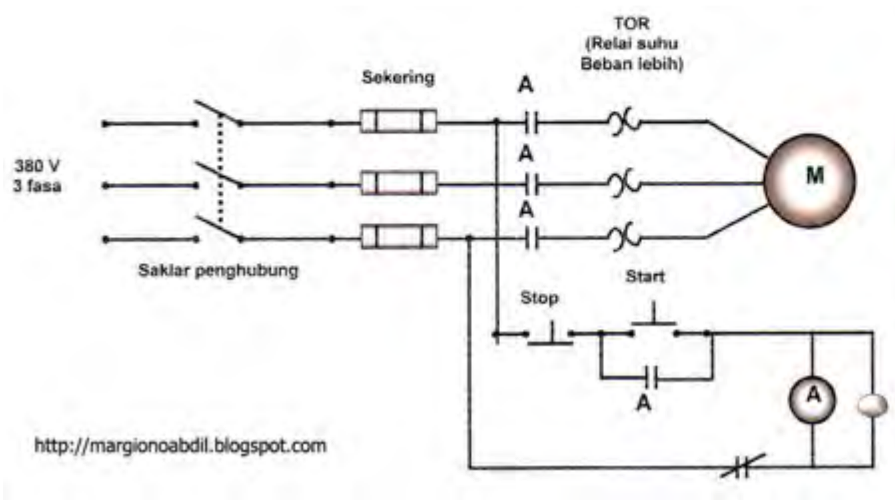
6. Diagram Kelistrikan

Diagram Klistrikan menggambarkan suatu prinsip kerja dari suatu peralatan dengan melihat hubungan pengawatannya yang disertai dengan simbol-simbol, ukuran yang jelas serta dapat juga menunjukkan lokasi dimana peralatan tersebut dipasang.

Membaca dan menulis diagram kelistrikan ada dua cara yaitu Penulisan secara Vertikal dan Penulisan secara Horizontal.



Gambar 4 Diagram kelistrikan secara vertikal



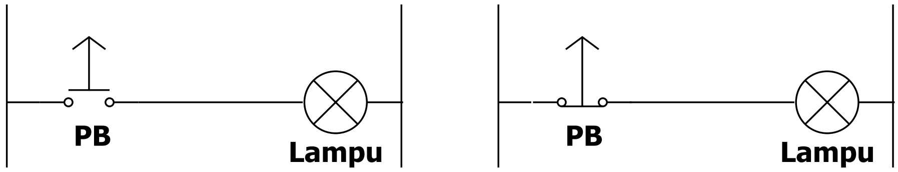
Gambar 5 Diagram kelistrikan secara horizontal

7. Rangkaian Dasar Kelistrikan

Ada beberapa rangkaian dasar rangkaian listrik yang umum dipakai oleh para teknisi atau perancang rangkaian rangkaian listrik.

a. Rangkaian ON / OFF

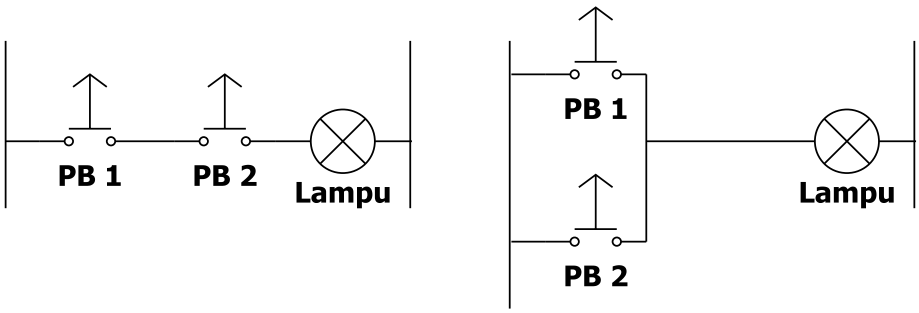
Rangkaian ini adalah yang paling dasar sekali dalam pengoperasian power on atau power off (switch off atau switch on).



Gambar 6 Rangkaian ON / OFF.

b. Rangkaian AND dan OR

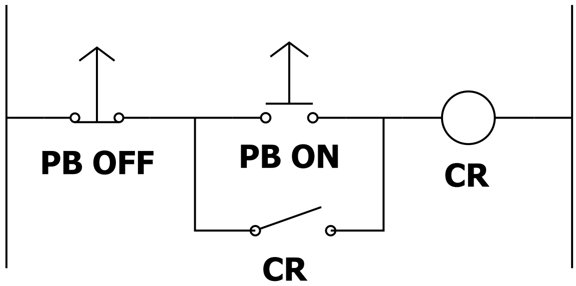
Rangkaian dasar kontrol yang paling sederhana adalah rangkaian rangkaian seri (AND) dan rangkaian rangkaian paralel (OR).



Gambar7 Rangkaian AND dan OR.

c. Rangkaian Pengunci

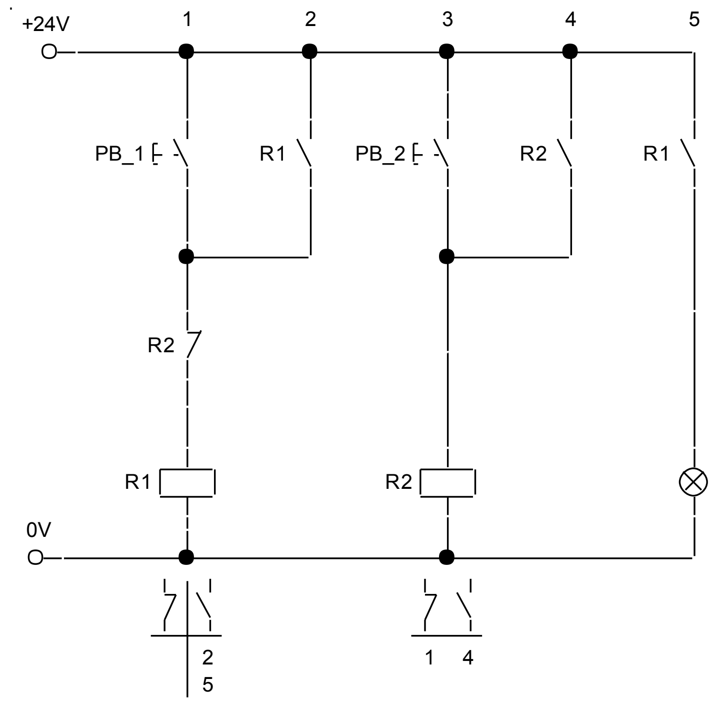
Jika tombol ON di tekan, akan menyebabkan coli relay (CR) aktif. Dengan aktifnya relay (CR) akan menyebabkan kontak relay (CR) terhubung, dengan terhubungnya kontak relay (CR) akan menyebabkan aliran listrik ke koil relay dipertahankan.



Gambar 8 Rangkaian pengunci.

d. Rangkaian Pengaman

Jika tombol PB_1 ditekan lampu akan menyala. Bila PB_2 ditekan lampu akan mati akibat diputus oleh NC R2.

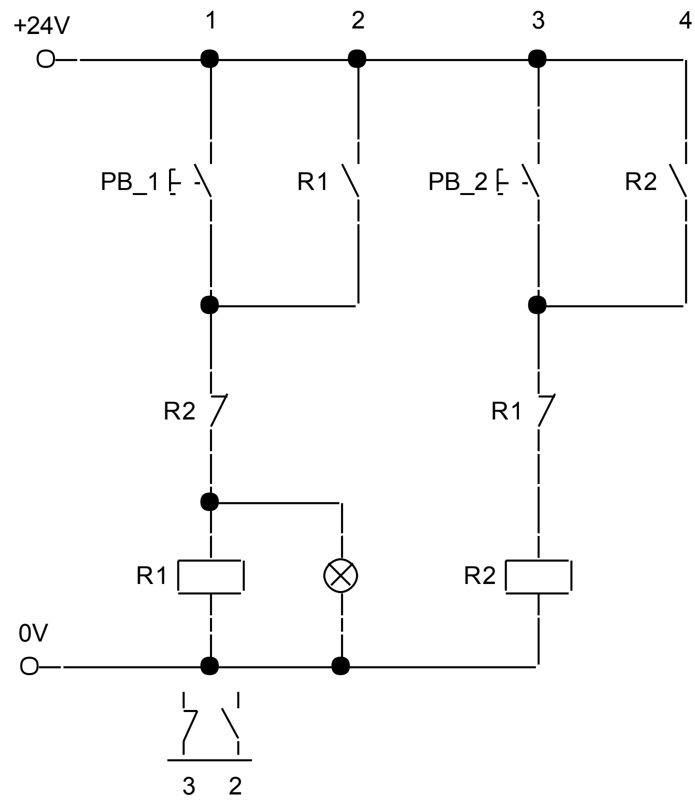


Gambar 9 Rangkaian pengaman

e. Rangkaian *Interlock* (bergantian)

Rangkaian ini berfungsi jika tombol PB_1 ditekan maka R1 akan aktif dan R2 tidak bisa aktif. Begitu juga sebaliknnya bilamana tombol PB_2 ditekan maka R2 akan aktif dan R2 tidak bisa aktif.

Dalam hal ini mana yang terlebih dahulu mengaktifkan (tombol PB_1 atau PB_2) akan terjadi balapan mana yang lebih dahulu menekan tombol.



Gambar 10 Rangkaian *interlock*

C. Tugas

Untuk memahami dan mendalami konsep dan pengertian kontrol, kerjakanlah tugas-tugas berikut ini :

1) Buatlah kelompok belajar, masing-masing kelompok maksimum 4 orang.
2) Diskusikan, buatlah kesimpulan tentang sistem kontrol.

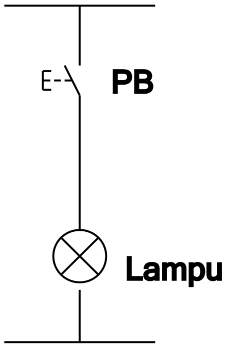
D. Tes Formatif

1. Apakah yang dimaksud dengan kontrol ?
2. Apa perbedaan *Feedback* Kontrol dengan sekuensi kontrol ?
3. Ada berapa macam tipe kontrol, sebutkan dan jelaskan !
4. Sebutkan perbedaan kontrol menggunakan relay dengan kontrol yang terbuat komponen semikonduktor !
5. Gambarkan rangkaian kontrol listrik dasar !
6. Gambarkan rangkaian pengunci !
7. Gambarkan rangkaian *interlock* !

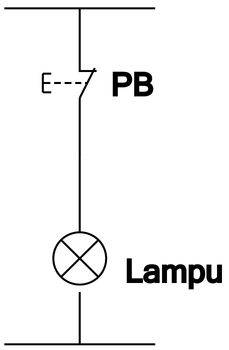
E. Lembar Kerja

Langkah Kerja

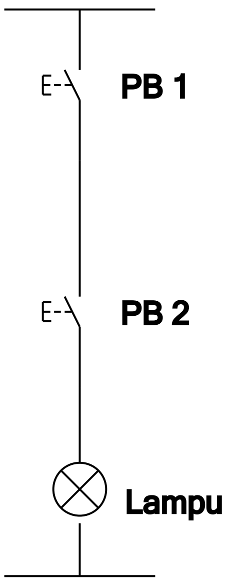
1. Rangkaian rangkaian ON dibawah ini !



2. Tekan tombol PB, apa yang terjadi pada lampu ....
3. Rangkai rangkaian OFF dibawah ini !



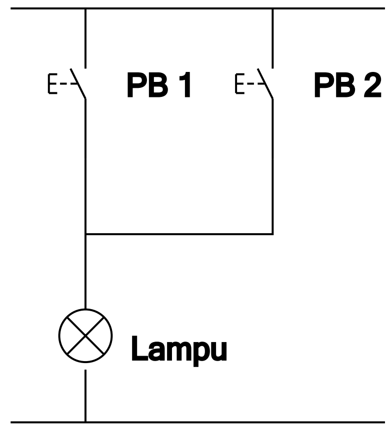
4. Tekan tombol PB, apa yang terjadi pada lampu ....
5. Rangkai rangkaian AND dibawah ini !



6. Isilah tabel kebenaran dibawah ini berdasarkan gambar pada soal nomor 3 dan instruksi-instruksi dalam tabel !

PB 1	PB2	Lampu
OFF	OFF	
ON	OFF	
OFF	ON	
ON	ON	

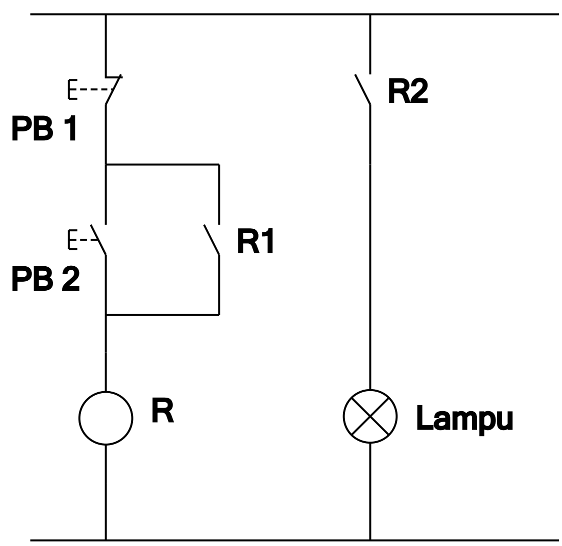
7. Rangkai rangkaian AND dibawah ini !



8. Isilah tabel kebenaran dibawah ini berdasarkan gambar pada soal nomor 7 dan instruksi-instruksi dalam tabel !

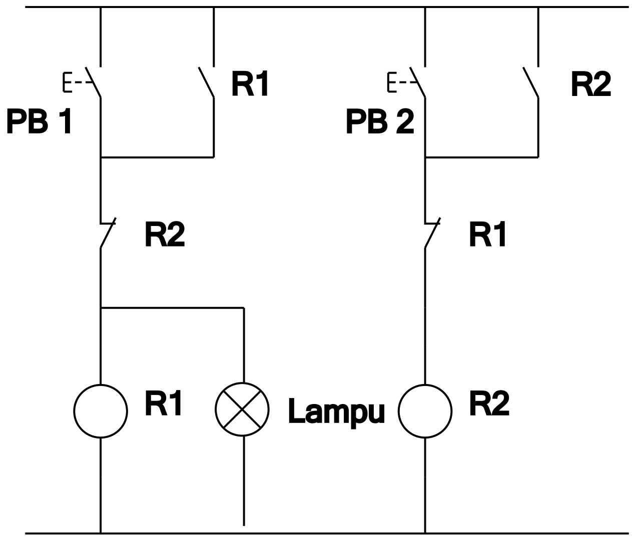
PB 1	PB2	Lampu
OFF	OFF	
ON	OFF	
OFF	ON	
ON	ON	

9. Rangkai rangkaian pengunci dibawah ini !



10. Tekan tombol PB 1, apa yang terjadi pada lampu ?
11. Tekan tombol PB 2, apa yang terjadi pada lampu ?

12. Rangkai rangkaian pengaman dibawah ini !



13. Tekan salah satu tombol, apa yang terjadi ?

BAB 2

KEGIATAN BELAJAR 2 :
PROGRAMMABLE LOGIC CONTROLLER

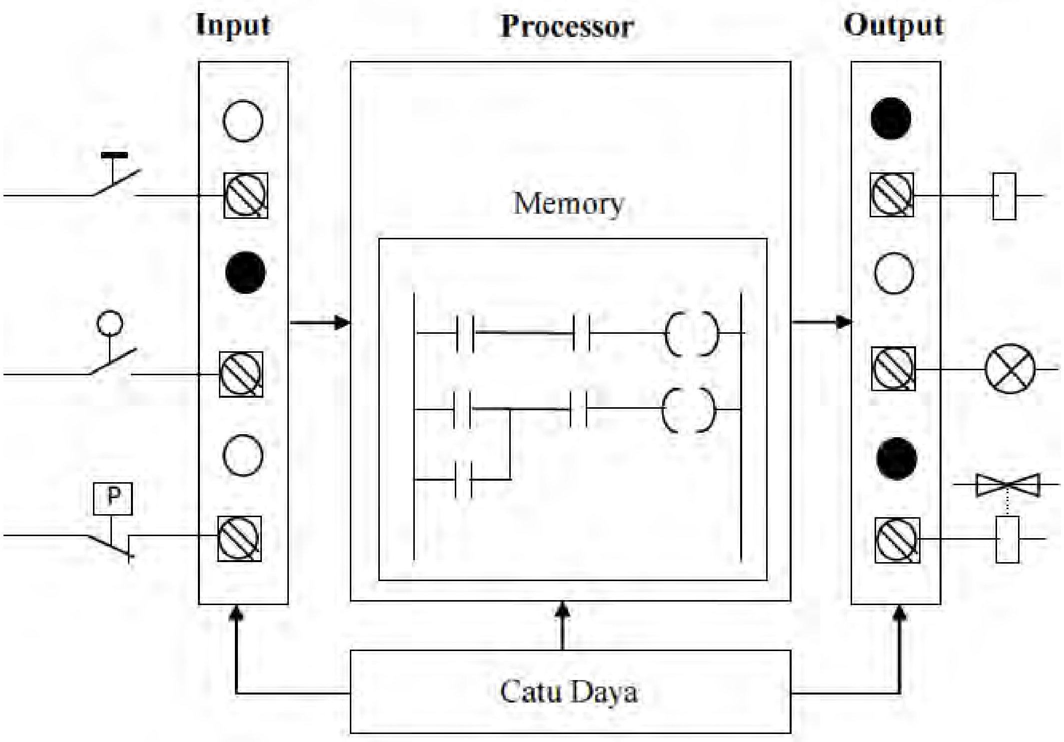
A. Tujuan Pembelejaran

Setelah menyelesaikan kegiatan belajar 2, siswa diharapkan mampu :

1. Memahami dan mengaplikasikan pengertian PLC.
2. Memahami dan mengaplikasikan blok diagram PLC.
3. Memahami dan mengaplikasikan pemrograman PLC.

B. Uraian Materi

Programmable Logic Controller (PLC) yang berfungsi sebagai pengendali yang perilakunya dapat disesuaikan dengan kebutuhan pengguna, serta penyusunan program kontrolnya berdasarkan pada suatu rangkaian kelistrikan yang diaplikasikan kedalam pernyataan logika (logic). Dengan cara memasukan program kedalam input data yang ada pada Programmable Logic Controller (PLC) melalui Programming Console dan Programming Ladder melalui komputer (PC).



Gambar 11 Diagram sistem kontrol Programmable Logic Controller (PLC)

1. **Keunggulan PLC dibandingkan dengan konvensional kontrol panel**

A. Sistem kontrol Programmable Logic Controller (PLC)

1) Wiring relatif sedikit.
2) Spare part mudah.
3) Maintenance relatif mudah.
4) Pelacakan kesalahan sistem lebih sederhana.
5) Konsumsi daya relatif rendah.
6) Dokumentasi gambar sistem lebih sederhana dan mudah dimengerti.
7) Modifikasi sistem lebih sederhana dan cepat.

B. Sistem konvensional kontrol panel

1) Wiring relatif komplek.
2) Spare part relatif sulit.
3) Maintenance membutuhkan waktu yang lebih lama.
4) Pelacakan kesalahan sistem sangat komplek.
5) Konsumsi daya listrik relatif tinggi.
6) Dokumentasi gambar sistem lebih banyak.
7) Modifikasi sistem membutuhkan waktu yang lama.

2. Keuntungan menggunakan PLC

A. Lama pengeTjaan untuk sistem baru design ulang lebih singkat.
B. Modifikasi sistem mUngkin tanpa tambah biaya jika masih ada spare I/O.
C. Perkiraan biaya suatu sistem design baru lebih past.
D. Relatif mudah untuk dipelajari.
E. Design sistem baru mudah dimodifikasi.
F. Aplikasi PLC sangatlah leas.
G. Mudah dalam hal Maintenance.
H. Sangatlah handal.
I. Standarisasi sistem kontrol lebih mudah diterapkan
J. Lebih aman untuk teknisi.

3. Beberapa contoh aplikasi dengan PLC :

A. Sistem konveyor
B. Pengolahan air limbah
C. Lampu merah Lalulintas
D. Robot kontroi
E. Mesin Moulding (Moulding Injection)
F. Pabrik Semen
G. Pabrik Sepatu

H. Otomatisasi Bangunan

I. Kontrol Lift

J. Pabrik makanan

K. Pabrik rokok

L. Pabrik mobil

M. Pabrik keramik

N. Pompa bensin

O. Pabrik kaleng makanan

P. Mesin sablon

Q. Pabrik kaca

R. Pabrik beton bertulang

Piranti input umumnya menggunakan signal tegangan 24 VDC dan outputnya menggunakan relay atau menggunakan semikonduktor.

Perangkat Programmable Logic Controller (PLC) ada 2 macam yaitu :

1. Programmable Logic Controller (PLC) Fixed

   Adalah PLC yang dari konstruksinya sudah menjadi satu antara piranti input, output dan pemprosesnya menjadi satu bagian, tetapi umunya kapasitasnya kecil.

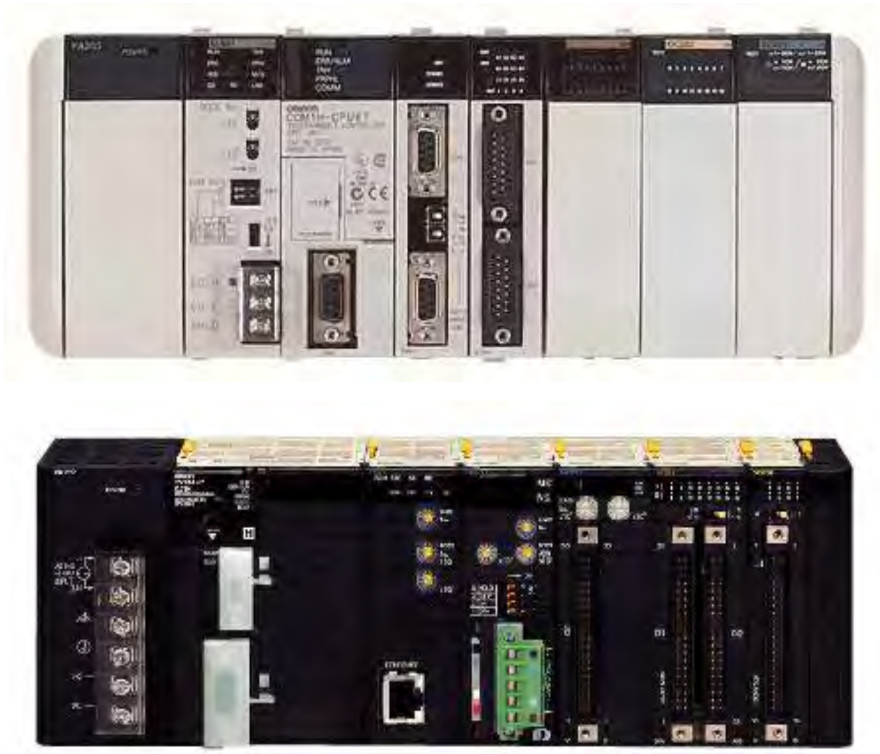


Gambar 12 Programmable Logic Controller (PLC) Fixed

2. Programmable Logic Controller (PLC) Modular

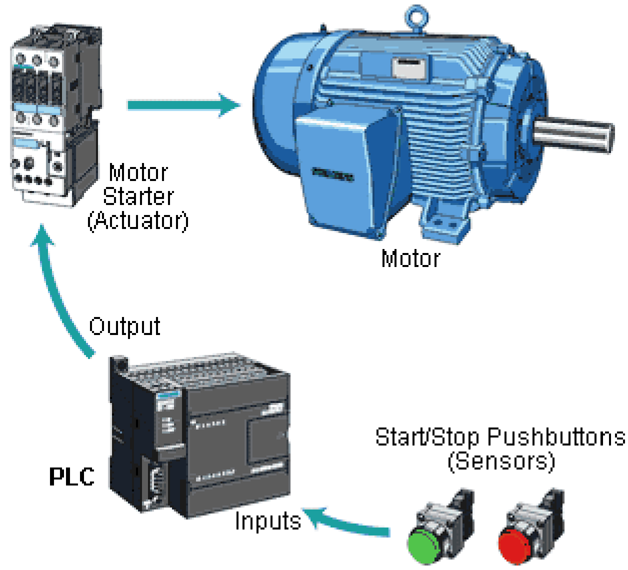
Adalah PLC yang dari konstruksinya terpisah seperti piranti input, output dan pemprosesnya menjadi bagian dari modul-modul. Umumnya kapasitasnya besar.

Ukuran kapasitas dan PLC ditentukan oleh merk, tipe dan model yang dibuat oleh pabrik.



Gambar 13 Programmable Logic Controller (PLC) Modular

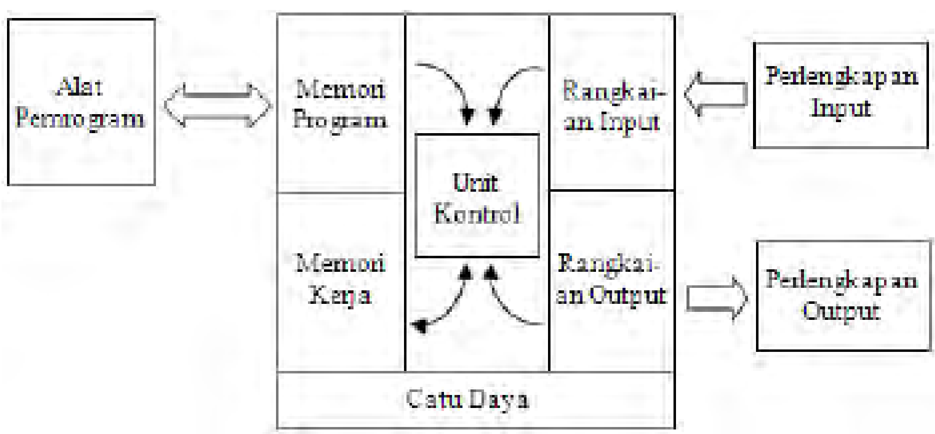
Programmable Logic Controller (PLC) tidak dapat dipasang langsung untuk piranti yang membutuhkan power tinggi seperti motor listrik karena output PLC kemampuannya terbatas. Oleh karena itu harus dipasang suatu perangkat penguat yang disebut buffer.



http://www.ustudy.in/node/210

Gambar 14 Output PLC menggunakan buffer.

4. Blok Diagram Programmable Logic Controller (PLC)



Gambar 15 Blok diagram Programmable Logic Controller (PLC)

A. Power Supply Unit (Catu Daya)

Unit ini berfungsi untuk memberikan sumber daya pada PLC. Modul ini sudah berupa Switching Power Supply.

B. CPU (Central Processing Unit)

Unit ini merupakan otak dari PLC. Disinilah program akan diolah sehingga sistem kontrol yang telah kita design bekerja seperti yang kita inginkan. CerlPU PLC Omron sangat bervariasi macamnya tergantung pada maasing-masing tipe PLC-nya.

C. Memory Unit

RAM : Random Acces Memory

EPROM : Erasable Programmable Read Only Memory

EEPROM : Electrical Erasable Programmable Read Only Memory

D. Input Unit

Digital Input : Input point digital

- AC / DC 24 Volt
- DC 5 Volt
- AC 110 / 220 Volt

Analog Input : Input point linier

- 0 – 10 Volt DC
- - 10 Volt – +10 Volt DC
- 4 – 20 mA DC

E. Output Unit

Digital Output : Output point digital

- Relay Output
- 110 Volt AC Output (Solid State)
- 220 Volt AC Output (Solid State)
- 24 Volt DC Output (type PNP / NPN)

Analog Output : Output point linier

- 0 – 10 Volt DC
- - 10 Volt – +10 Volt DC
- 4 – 20 mA DC

F. Peripheral

- ✓ Handheld Programming Console
- ✓ SSS : Sysmate Support Software
- ✓ CX Programmer Software
- ✓ LADSimp
- ✓ PROM Writer
- ✓ GPC : Graphic Programming Console
- ✓ FIT : Factory Intelegent Terminal

5. Sistematika Mendesign suatu Sistem Programmable Logic Controller (PLC)

A. Mempelajari sampai mengerti betul urutan kerja (sequence) sistem tersebut.

B. Membuat flowchart dari suatu sistem.

C. Membuat daftar semua input dan output terhadap I/O point dari PLC.

D. Menerjemahkan flowchart ke Ladder Diagram dan disesuaikan dengan daftar I/O yang telah anda buat.

E. Memeriksa program jika masih ada kesalahan logika disesuaikan dengan logika pada flowchart dan juga harus sesuai dengan daftar I/O point yang telah kita buat.

F. Mentransfer program ke memori PLC (Training Kit PLC).

G. Mensimulasikan program pada Training Kit PLC dan menganalisanya apakah sudah sesuai dengan yang kita

inginkan.

H. Jika simulasi sudah benar, barulah kita menghubungkan semua peralatan Input dan Output ke terminal PLC (pada aplikasi yang sesungguhnya).

I. Memeriksa kembali hubungan kabel dan peralatan Input dan Output ke PLC, setelah yakin sudah benar barulah kita melakukan testing program lagi.

J. Jika sistem sudah berjalan dengan baik dan benar, barulah dilakukan dokumentasi gambar sistem secara sistematis sehingga mudah dimengerti dan mudah dipelajari.

6. Memory Programmable Logic Controller (PLC)

Memory Programmable Logic Controller (PLC) terdiri dari :

A. IR (Internal Relay)

Internal relay mempunyai pembagian fungsi seperti IR input, IR output, dan juga IR work area (untuk pengolahan data pada program). IR input dan IR output adalah IR yang berhubungn dengan terminal input dan output pada PLC. Sedangkan IR work area tidak dihubungkan ke terminal PLC, akan tetapi berada dalam internal memory PLC dan fungsinya untuk pengolahan logika program kita (manipulasi program).

Ada juga IR yang difungsikan untuk SYSMAC BUS Area, Special I/O Unit Area, Optical I/O Unit Area, dan Group 2 High Density I/O Unit Area.

- **SYSMAC BUS Area** berfungsi untuk omunikasi data PLC antara CPU PLC dan //O Unit PLC hanya dengan menggunakan 2 kabel saja (RS 485), maksimum 200m.
- **Special I/O Unit Area** merupakan IR yang digunakan oleh Special I/O Unit PLC (contoh: Analog input, Analog output

dli) untuk mengatur, menyimpan dan mengolah datanya.

- **Optical I/O Unit Area** adalah IR yang digunakan untuk mengolah dan menyimpan data dari Optical I/0 Unit PLC.
- **Group 2 High Density I/O Unit Area** adalah IR untuk menyimpan dan mengolah data dari High density I/O unit group 2.

B. SR (Special Relay)

Special Relay adalah relay yang mempunyai fungsi-fungsi khusus seperti untuk flags. (misalnya pada instruksi penjumlahan terdapat keleblhan digit pada hasilnya [carry flag]), kontrol bit PLC, informasi kondisi PLC, dan system clock (pulsa I detik, 0,2 detik, dll.)

C. AR (Auxilary Relay)

Terdiri dari flags dan bit untuk tujuan-tujuan khusus. Dapat menunjukkan kondisi PLC yang disebabkan oleh kegagalan sumber tegangan, kondisi Special I/O, kondisi input/output unit, kondisi CPU PLC, kondisi memori PLC, dll.

D. HR (Holding Relay)

Dapat difungsikan untuk menyimpan data (bit-bit panting) karena tidak akan hilang walaupun sumber tegangan PLC mati.

E. LR (Link Relay)

Digunakan untuk data link pada PLC Link System. Artinya untuk tukar menukar informasi antar dua PLC atau lebih dalam suatu sistem kontrol yang sating berhubungan satu dengan yang lain dan menggunakan banyak PLC (minimum 2 PLC).

F. TR (Temporary Relay)

Berfungsi untuk menyimpan sementara kondisi logika pada program pada ladder diagram yang mempunyai titik percabangan khusus.

G. TC (Timer / Counter)

Untuk mendefinisikan suatu sistem waktu tunda /time delay [TIMER] ataupun untuk penghitung [COUNTER]. Untuk Timer mempunyai orde 100 ms, ada yang mempunyai orde 10 ms yaitu TIMH (15). Untuk TIM 000 s/d TIM 015 dapat dioperasikan secara interupt untuk mendapatkan waktu yang lebih presisi.

H. DM (Data Memory)

Data memory berfungsi untuk penyimpanan data-data pogrim karena isi DM tidak akan hilang (reset) walaupun sumber tegangan PLC mati.

Macam-macam DM adalah sebagai berikut :

✓ **DM Read/Write :**

Pada DM ini bisa dihapus dan ditulis oleh program yang kita buat. Jadi sangat berguna unutk manipulasi data program.

✓ **DM Special I/O unit :**

DM ini berfungsi untuk menyimpan dan mengolah hasil dari Special I/O Unit, mengatur dan mendefinisikan sistem kerja Special I/O Unit.

✓ **DM History Log :**

Pada DM disimpan informasi-informasi penting pada saat PLC terjadi kegagalan sistem operasionainya. Pesan-pesan kesalahan sistem PLC yang disimpan adalah berupa kode-kode angka tertentu.

- ✓ **DM Link Test Area :**

  Berfungsi untuk menyimpan informasi-informasi yang menunjukkan status dari Sistem Link PLC.

- ✓ **DM Setup :**

  Berfungsi untuk Setup kondisi default (kondisi kerja saat PLC aktif)_ Pada DM inilah kemampuan kerja suatu PLC didefinisikan untuk per Lama kalinya sebelum PLC tersebut diprogram dan dioperasikan pada suatu sistem kontrol. Tentu saja setup PLC tersebut disesuaikan dengan sistem kontrol yang bersangkutan.

I. UM (Upper Memory)

Memori ini berfungsi untuk menyimpan dan menjalankan program kita (user program). Kapasitasnya tergantung pada masing-masing tipe PLC yang dipakai.

Semua memori (selain DM dan UM) di atas dapat anda bayangkan seperti relay yang mempunyai coil, contact NO dan contact NC. Timer/Counter juga dapat dibayangkan seperti Timer/Counter pada umumnya. Timer/Counter pada PLC juga mempunyai NO dan NC.

DM tidak mempunyai contact, yang ada hanyalah channel/word saja. DM dapat difungsikan untuk menyimpan data-data penting yang tidak boleh hilang waktu power padam, atau untuk manipulasi program kita.

Memori yang sifatnya dapat menyimpan data program jika listrik mati adalah DM dan HR. sedangkan yang lain akan kembali reset (hilang).

7. Bahasa Pemrograman PLC

Bahasa pemrograman pada PLC pada dasarnya merupakan bentuk dari berbagai informasi yang dibutuhkan untuk mengontrol dan

memonitor suatu proses. Bahasa pemrograman ini merupakan komposisi dari satu set instruksi yang mengikuti aturan-aturan sintaksis yang tepat dalam menetapkan metode penulisan, pembacaan dan modifikasi suatu program kontrol. Jadi istilah „bahasa pemrograman" mengacu pada cara yang digunakan oleh programer untuk berkomunikasi dengan PLC.

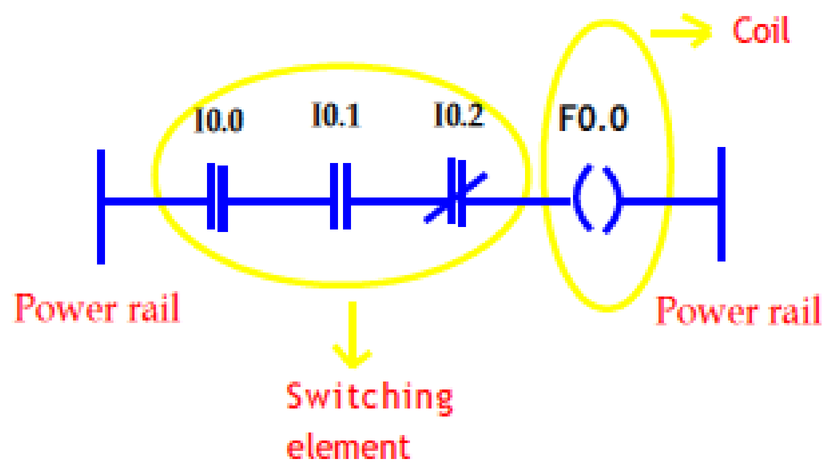
Tergantung pada pabrikan PLC, Setiap jenis PLC hanya dapat diprogram dengan bahasa pemrograman tertentu. Ada beberapa jenis PLC yang dapat diprogram dengan berbagai bahasa pemrograman sesuai standard IEC. Tetapi ada pula PLC yang hanya dapat diprogram dengan satu jenis bahasa (misalnya Ladder Diagram).

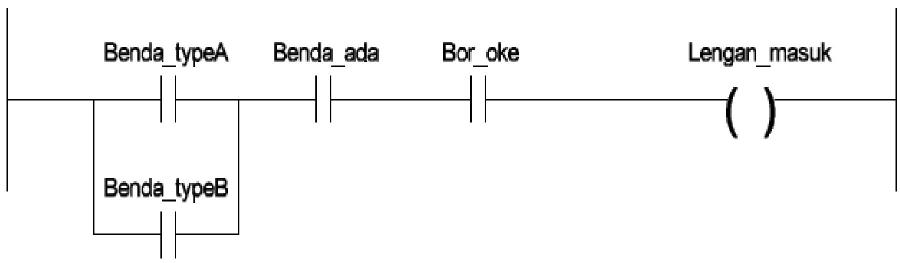
IEC atau **International Electrotechnical Commission** adalah suatu standardisasi internasional nirlaba yang menyiapkan dan mempublikasi kan standar internasional untuk semua teknologi elektrik, elektronika, dan teknologi lain yang terkait, yang secara kolektif dikenal dengan "elektroteknologi". Standar IEC meliputi berbagai teknologi dari pembangkitan, transmisi, dan distribusi listrik hingga perlengkapan rumah tangga dan perlengkapan kantor, semikonduktor, serat optik, baterai, tenaga surya, nanoteknologi dan tenaga air laut, serta berbagai hal lain. IEC juga mengelola skema penilaian kesesuaian yang menyatakan apakah suatu perangkat, sistem, atau komponen sesuai dengan standar internasional. IEC menerbitkan standar bersama dengan IEEE dan mengembangkan standar-standar bersama dengan ISO dan juga ITU.Instruksi IEC mempunyai format standar yang dikenal oleh beragam jenis PLC.

Komisi Elektroteknik Internasional (IEC) mengembangkan standar IEC 1131 dalam upaya untuk membakukan programmable controller. Salah satu tujuan komite ini adalah untuk menciptakan seperangkat instruksi PLC yang dapat digunakan dalam semua PLC. Meskipun standar IEC 1131 mencapai status standar internasional pada bulan Agustus 1992, upaya untuk menciptakan standar PLC global telah menjadi tugas yang sangat sulit untuk dicapai, akibat keragaman produsen PLC dan masalah ketidakcocokan antar merk PLC. Namun, terobosan yang telah telah dibuat sejauh ini telah berdampak besar pada cara PLC akan diprogram di masa depan.

Menurut IEC 1131-3. ada 5 jenis bahasa pemrograman PLC, yaitu :

- ✓ **Ladder Diagram Language (LAD),** yaitu bahasa pemrograman PLC yang berbasis relai ladder logic diagram atau bahasa pemrograman yang ditulis secara grafikal.





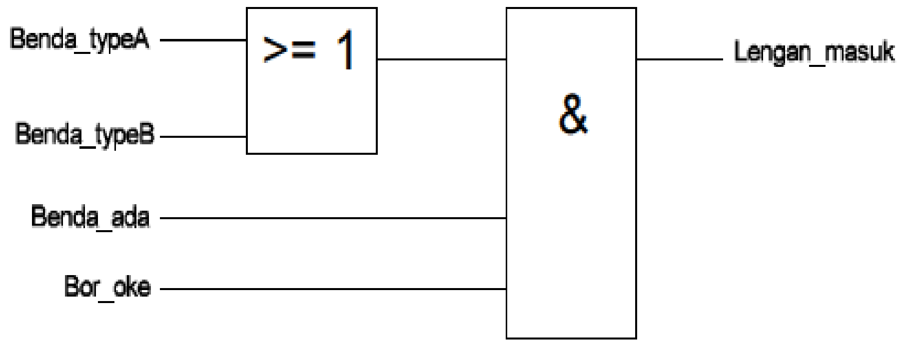
Gambar 16 Pemrograman dengan Ladder Diagram.

Ladder diagram adalah sebuah bahasa pemrograman gambar diturunkan dari diagram rangkaian pengawatan kontrol relai secara langsung. Program pada PLC disebut program Ladder karena bentuknya yang mirip tangga. Ladder diagram terdiri dari susunan kontak-kontak yang disusun dari sebelah kiri ke kanan pada diagram; kontak-kontak ini disambungkan ke elemen-elemen pensakelaran (kontak NO/NC) melalui jalur arus dan elemen koil.

Ladder Diagram mempunyai bentuk seperti rangkaian listrik. Sebuah Ladder diagram terdiri dari power rail pada sisi kanan dan kiri diagram, dihubungkan dengan rung oleh *switching element* dan *coil element* tertentu.

- ✓ **Function Block Diagram Language (FBD),** yaitu bahsa pemrograman yang berbasis block-block grafikal mengac pada blok-blok diagram yang digunakan pada aljabar Boolean.

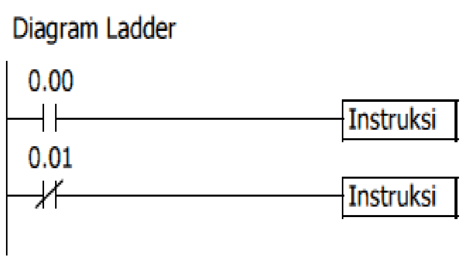
  Pada FBD, fungsi dan blok fungsi digambarkan dengan grafik dan dihubungkan melalui jaringan. FBD berasal dari *logic diagram* pada sirkit elektronik.



Gambar 17 Pemrograman dengan Function Blok Diagram.

Dalam diagram blok fungsi, fungsi-fungsi dan blok fungsi digambarkan secara grafik dan dihubungkan ke dalam jaringan. Diagram blok fungsi berasal dari diagram logika untuk desain rangkaian-rangkaian elektronik.

- ✓ **Statement List Language (STL),** yaitu bahasa pemrograman yang berbasis bahasa kode seperti bahasa assembler atau bahasa pemrograman yang dituliskan secara tekstual.
  Daftar kalimat (statement list) adalah sebuah bahasa kalimat jenis assembler bercirikan model mesin sederhana (prosesor hanya dengan satu register).



Mneumonik

Alamat	Instruksi	Operand
00000	LD	0.00
00001	Instruksi	
00002	LD NOT	0.01
00003	Instruksi	

Gambar 18 Pemrograman dengan Statement List.

Daftar instruksi diformulasikan dari instruksi kontrol yang berisi sebuah operator (pengerja) dan sebuah operand (yang dikerjakan). Berikut ini Contoh Bahasa Daftar Instruksi

Berkenaan dengan filosofi bahasa, ladder diagram, diagram blok fungsi dan daftar instruksi telah ditetapkan bagaimana cara menggunakannya dengan PLC saat ini. Bahasa-bahasa ini bagaiamanapun dibatasi untuk fungsi-fungsi dasar dengan memperhatikan elemen-elemennya. Perbedaan diantaranya dikarenakan oleh pabrik pembuatnya. Keunggulan bahasa-bahasa ini tetap dipertahankan terutama dalam penggunaan fungsi-fungsi dan blok-blok fungsi.

✓ **Structured Test Language (ST),** yaitu bahasa pemrograman yang berbasis bahasa pascal dengan, sangat prosedural, menggunakan loop statement dan kondisional atau secara tekstual.

Teks terstruktur adalah bahasa tingkat-tinggi yang berbasis Pascal, terdiri dari ekspresi-ekspresi dan instruksi-instruksi. Instruksi-instruksi secara pokok dapat dikategorikan menjadi:

- Instruksi-instruksi pilihan seperti: IF, THEN, ELSE, dll.
- Instruksi-instruksi pengulangan seperti: FOR, WHILE, dll dan
- Blok fungsi harapan/hasil.

Berikut merupakan contoh bahasa teks terstruktur

Contoh 1.

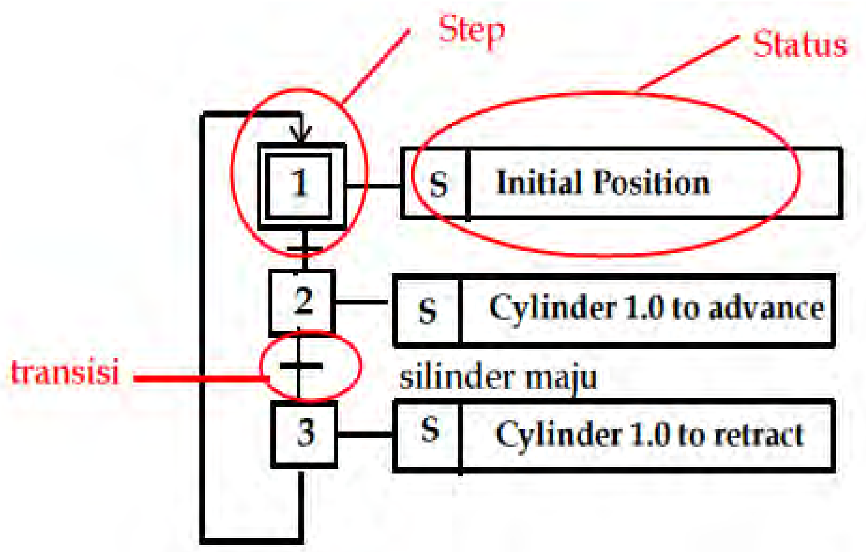
Lengan_masuk = (Benda_typeA **OR** Benda_typeB) **AND** Benda_ada **AND** Bor_oke;

Contoh 2.

Sleeve_on:=(Part_TypeA OR Part_TypeB) AND Part_present AND Drill_OK

Teks terstruktur memungkinkan aplikasi yang banyak, melebihi fungsi teknologi secara murni, seperti problem-problem algoritma (kontrol algoritma tingkat tinggi) dan penanganan data (analisa data maupun pemrosesan struktur data yang kompleks).

- ✓ **Sequential Function Chart (SFC),** yaitu bahasa pemrograman berbasis bahasa grafikal berdasarkan alur program (flowchart) Meskipun secara fungsi dan struktur dari bahasa-bahasa ini sangat berbeda, mereka dikategorikan sebagai satu keluarga bahasa oleh IEC 1131-3 dengan pelengkapan elemen-elemen struktur (pernyataan variabel, bagian-bagian organisasi seperti halnya fungsi dan blok fungsi, dll) dan elemen-elemen konfigurasi. *Dari kelima bahasa program diatas yang sering digunakan adalah: Ladder Diagram (LAD), Diagram Blok Fungsi (Function Block Diagram/FBD) dan Daftar Instruksi (Statement List /STL).* Bahasa-bahasa tersebut dapat dikombinasi dalam banyak hal di dalam sebuah proyek PLC. Ketentuan telah dibuat untuk pengembangan lebih lanjut, (sebagaimana prinsip blok fungsi atau bahasa teks terstruktur) disamping detail informasi teknologi yang diperlukan (jenis data, dll).



Gambar 19 Pemrograman dengan Function Chart.

Chart fungsi urutan adalah resource bahasa untuk penstrukturan programprogram kontrol berorientasi urutan. Elemen-elemen dari chart fungsi urutan adalah langkah-langkah (step),pemindahan –pemindahan (transisi),cabang alternatif dan percabangan paralel.

Setiap step menampilkan status pemrosesan dari program kontrol, mana yang aktif dan tidak aktif. Step terdiri dari aksi-aksi yang maupun transisi yang diformulasikan dalam bahasa-bahasa standart IEC 1131-3. Setiap aksi dapat juga terdiri dari struktur-struktur berurutan. Keistimewaan ini memungkinkan tingkatan struktur dari program kontrol. Chart fungsi urutan merupakan sebuah alat yang unggul untuk desain dan penstrukturan program kontrol.

SFC merupakan language resource untuk membentuk sequence oriented control program. Elemen dari SFC meliputi step, transition, alternative dan parallel branching. Tiap step

menunjukkan status yang diproses pada control program, baik dalam keadaan aktif atau tidak.

C. Tugas

Untuk memahami dan mendalami konsep dan pengertian kontrol, kerjakanlah tugas-tugas berikut :

1. Apa kepanjangan dari PLC ?
2. Apa yang dimaksud dengan PLC ?
3. Buat blok diagram dari PLC !
4. Ada berapa jenis PLC yang berkembang di pasaran ?
5. Apa keuntungan PLC disbanding dengan relay control ?
6. Identifikasi PLC yang ada di sekolahmu !

D. Tes Formatif

PETUNJUK KHUSUS :

Pilihlahsalahsatujawaban yang andaanggap paling benar !

1. Sebutkan dua macam bentuk program kendali PLC:
   a. Program diagram ladder dan blok fungsi
   b. Program diagram ladder dan program mneumonik
   c. Program diagram ladder dan program teks terstruktur
   d. Program blok fungsi dan program mneumonik
   e. Program blok fungsi dan teks terstruktur

2. Bahasa Pemprograman (statement list) pada PLC disebut juga ?
    a. Ladder Diagram
    b. Alur Diagram
    c. Mnemonic Code
    d. Data Base
    e. Turbo Pascal

3. Digram ladder terdiri atas sebuah garis vertikal di sebelah kiri yang disebut ?
    a. Rung
    b. Diagram
    c. Instruksi
    d. Bus Bar
    e. Kontak

4. Eksekusi Program semata-mata merupakan salah satu tugas yang dilakukan oleh PLC, hal tersebut sebagai bagian dari :
    a. Pemrograman
    b. Pengecekan Kondisi
    c. Operand Instruksi
    d. Waktu Siklus
    e. Mengeksekusi semua Intsruksi

5. Contoh bahasa pemrograman PLC dengan instruksi adalah ….
    a. Mnemonic code
    b. Diagram I/O
    c. Lader diagram
    d. Memori diagram

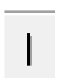
e. Logik diagram

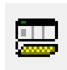
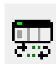
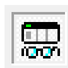
D. *Lembar Kerja*

CARA MEMBUAT PROGRAM SEDERHANA DENGAN PLC

a) PENDAHULUAN

Simbol dasar dan fungsi icon :

Simbol Icon	Fungsi Icon
	Kontak NO
	Kontak NC
	Garis Vertikal
	Garis Horizontal
	Output NO
	Output NC
	Menggambarkan fungsi-fungsi, misalkan : Timer, Counter dan mengakhiri program
	Untuk memilih
	Compile PLC Program
	On line, menghubungkan PLC dengan computer
	Down load, transfer program dari PC ke PLC
	Up load, transfer program dari PLC ke PC

Simbol Icon	Fungsi Icon
	Program Mode, kondisi membuat program
	Run, kondisi untuk mencoba program
	Monitor Mode, melihat kondisi kerja rangkaian

b) ALAT DAN BAHAN

a) Trainer PLC

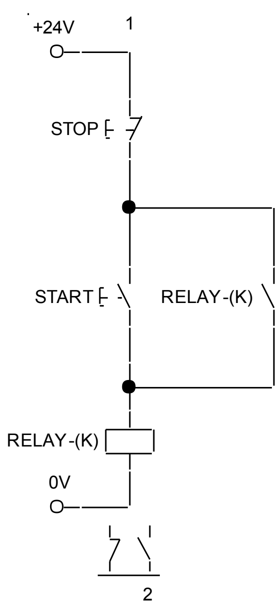
b) PC Komputer / Laptop

c) Kabel Jamper

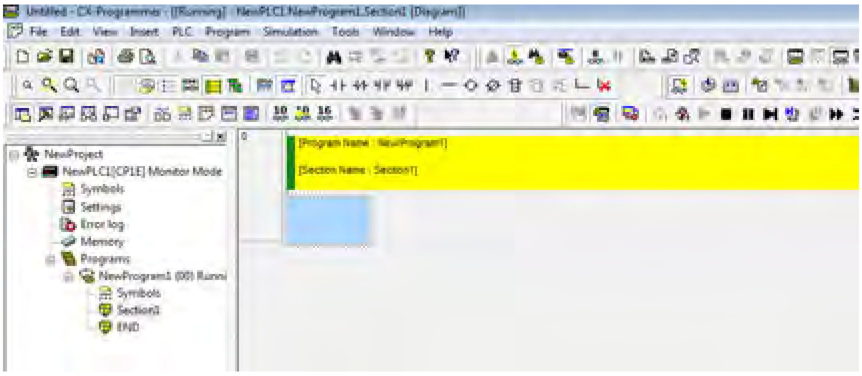
d) Obeng

c) LANGKAH KERJA

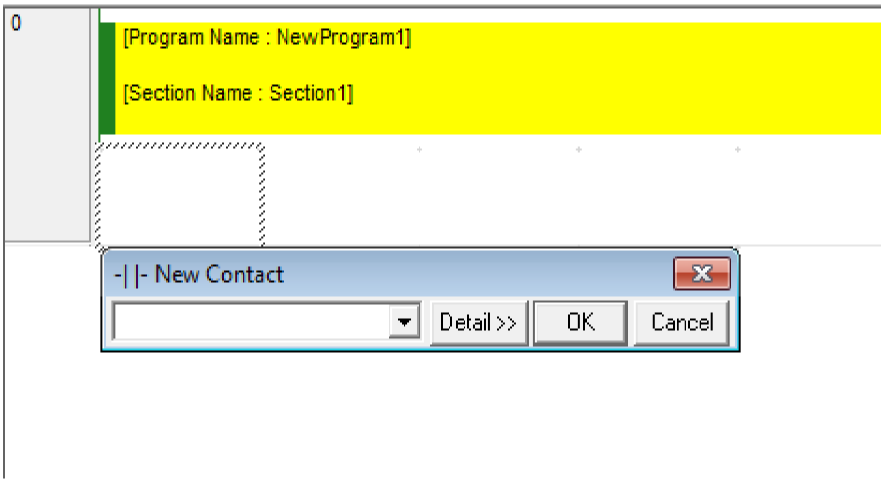
a) Buat ladder diagram dari gambar rangkaian listrik dibawah !



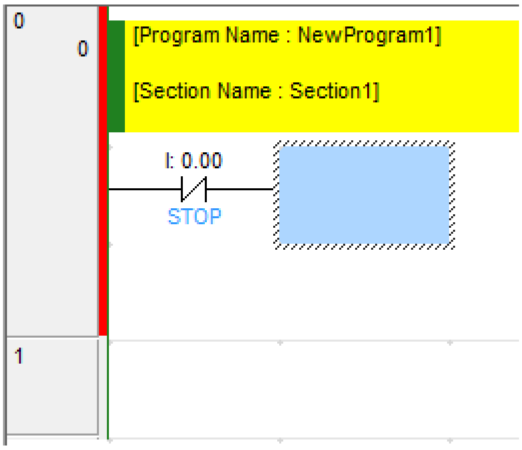
b) Dari lembar kerja CX Programer letakan cursor seperti pada gambar dibawah.



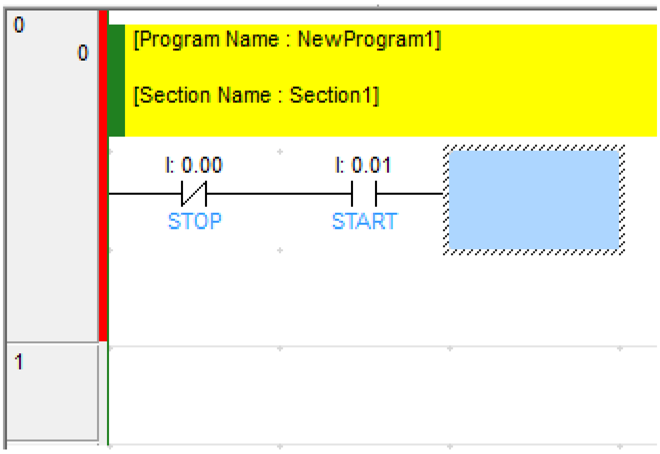
c) Klik kontak NC dan tempatkan pada blok biru dibawahnya sampai muncul kotak dialog seperti dibawah ini.



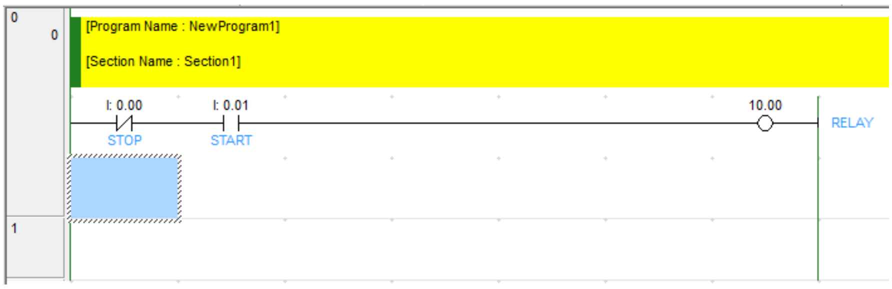
d) Isi alamat input 0.00 pada kotak lalu klik OK, isi juga kotak Edit Comment dan ketik STOP lalu klik OK hingga muncul gambar seperti dibawah ini.



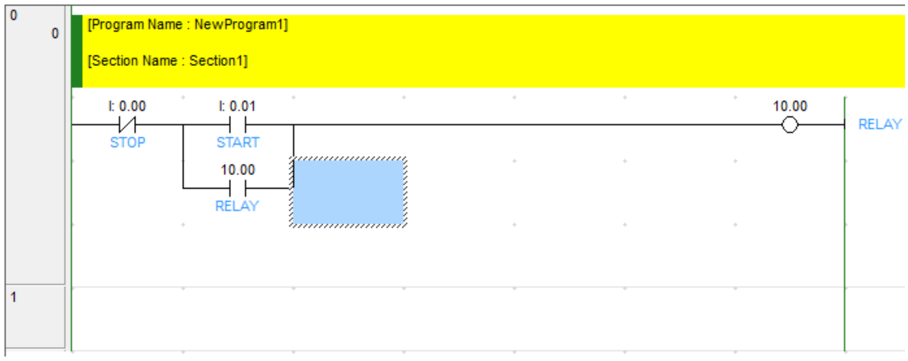
e) Ulangi langkah c) dan d) untuk membuat kontak NO dengan alamat 0.01 dan tulisan STOP hingga muncul gambar seperti dibawah ini.



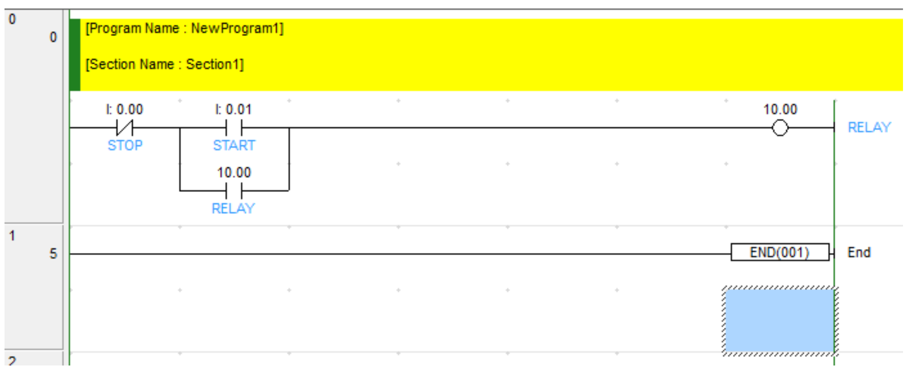
f) Ulangi langkah c) dan d) untuk membuat output relay dengan alamat 10.00 dan tulisan RELAY hingga muncul gambar seperti dibawah ini.



g) Untuk membuat pegunci, Ulangi langkah c) dan d) untuk membuat kontak NO dengan alamat 10.00 dan dibawah START dengan menggunakan kontak bantu  hingga muncul gambar seperti dibawah ini.



h) Tutup program dengan klik  kemudian isi kotak dialognya dengan END lalu klik OK seperti gambar dibawah, selesailah program.



d) KESELAMATAN DAN KESELAMATAN KERJA

- ✓ Gunakanlah pakaian praktik.
- ✓ Bacalah dengan seksama dan benar petunjuk praktikum.
- ✓ Hati-hati dengan aliran arus listrik.
- ✓ Jangan meletakkan peralatan di tepi meja.
- ✓ Kabel penghubung yang tidak terpakai jangan dekat dengan rangkaian.
- ✓ Tanyakan kepada instruktur hal-hal yang meragukan.

BAB 3

KEGIATAN BELAJAR 3 :
INSTRUKSI PROGRAM KONTROL PLC

A. Tujuan Pembelejaran

Setelah menyelesaikan kegiatan belajar 3, siswa diharapkan mampu :

1. Mampu mendefinisikan dan memahami susunan program PLC.
2. Mampu mendefinisikan dan memahami instruksi dan simbol PLC.
3. Mampu mendefinisikan dan memahami perakitan PLC dalam sistem aplikasi.
4. Mampu mendefinisikan dan memahami pengoperasian sistem PLC.

B. Uraian Materi

1. Pendahuluan

Pemrograman PLC terdiri dari instruksi-instruksi dasar PLC yang berbentuk logika pengendalian sistem kendali yang diinginkan.

Bahasa pemrograman biasanya telah disesuaikan dengan ketentuan dari pembuat PLC itu sendiri.

Dalam hal ini setiap pembuat PLC memberikan aturan-aturan tertentu yang sudah disesuaikan dengan programmer CPU yang digunakan PLC.

Manufactur atau pembuat PLC diantaranya sebagai berikut:

a. Allen Bradley (www.ab.com) -> Nama-nama PLC nya: Control Logix,PLC-5, SLC, Flex Logix, dll. sedangkan software yang dipakai adalah RSLogix dan RSLinx. http://www.ab.com/programmablecontrol/
b. SchneiderElectric.(http://www.telemecanique.com/en/functions_discover y/function_5_11.htm) -> Modicon Quantum, Compact, Momentum, Micro, Premium, dll. Software yang di pakai adalah Concept buat Modicon Quantum, dan ada lagi yang lain.
c. Siemens -> S7-400, S7-300, S5. Software yang dipakai Step7 (S7-400 dan S7-300) dan Step5 (buat S5, masih under DOS tampilannya).
d. OMRON -> CPM1A, CPM2A,CQM1H,CS1D-H,CS1D-S,CS1G-H,CS1H, CS1H-H ,CV1000,dll. Software yang dipakai adalah Syswin dan CX-Programmer.
e. Mitsubishi.
f. GE Fanuc.
g. Wago.
h. Hollysys dll.

2. Tipe Bahasa Pemrograman

Program PLC dapat dibuat dengan menggunakan beberapa cara yang disebut bahasa pemrograman. Bentuk program berbeda-

beda sesuai dengan bahasa pemrograman yang digunakan. Bahasa pemrograman tersebut antara lain :

diagram ladder, kode mneumonik, diagram blok fungsi, dan teks terstruktur seperti yang sudah dibahas pada Bab 1.

Beberapa merk PLC hanya mengembangkan program diagram ladder dan kode mneumonik.

3. Format Ladder Diagram

Format Ladder diagram terdiri dari beberapa baris koneksi. Setiap satu baris koneksi dari catu sebelah kiri ke kanan disebut *rung.* Sebuah program ladder dapat terdiri atas banyak *rung,*tergantung dari kompleksitas logika kontrol yang hendak diterapkan. Banyaknya rung maksimal yang dapat ditangani oleh sebuah PLC tergantung dari kapasitas memori untuk program yang dimilikinya. Setiap rung akan dieksekusi satu demi satu ,dari atas ke bawah.

Lamanya waktu eksekusi juga bergantung pada kompleksitas dan panjangnya rung.

Pemrograman dengan menggunakan editor Ladder ini membuat program yang menyerupai diagram pengkabelan kelistrikan ("electrical wiring diagram"). Diagram Ladder ini mungkin merupakan pilihan yang paling banyak dipakai dan disukai oleh banyak progamer PLC dan bagian perawatan karena menampilkan program dalam bentuk yang sudah dikenal dan mudah dianalisa. Diagram Ladder ini secara umum akan menggambarkan aliran daya dari arus listrik yang melalui sederetan kondisi logika input yang pada akhirnya akan mengaktifkan suatu output. Kumpulan Logika ini biasanya dibagi-bagi dalam beberapa baigian yang disebut "rung" atau

“network” untuk lebih memudahkan pemahaman dan analisa. CPU akan mengeksekusi rung atau network ini satu demi satu dimulai dengan dari kiri-ke-kanan dan kemudian dari atas-ke-bawah. Setelah CPU mengeksekusi network yang terakhir ia akan kembali kenetwork yang pertama

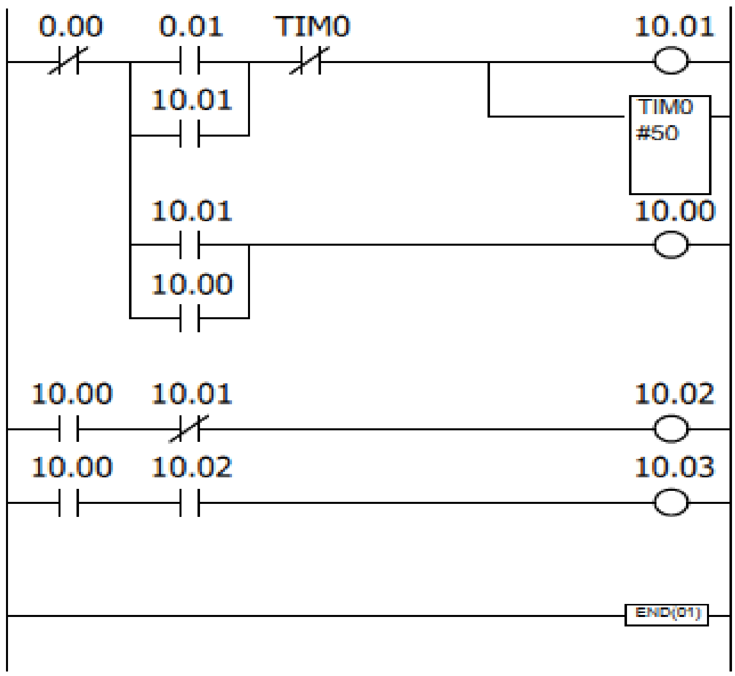
4. Pemrograman Ladder

Pemrograman Ladder Adalah bahasa pemrograman yang yang dibuat dari persamaan fungsi logika dan fungsi-fungsi lain berupa pemrosesan data atau fungsi waktu dan pencacahan. Ladder diagram terdiri dari susunan kontak- kontak dalam satu group perintah secara horizontal dari kiri ke kanan, dan terdiri dari banyak group perintah secara verikal. Contoh dari Ladder Diagram ini adalah: kontak normaly open, kontak normaly close, output coil, pemindahan data. Garis vertikal paling kiri dan paling kanan diasumsikan sebagai fungsi tegangan, bila fungsi dari group perintah . Sepanjang garis instruksi, ditempatkan kontak-kontak yang mengendalikan/mengkondisikan instruksi lain di sebelah kanan. Kombinasi logika kontak-kontak ini menentukan kapan dan bagaimana instruksi di sebelah kanan dieksekusi. Ada tiga bentuk utama dalam diagram Ladder sebagai berikut :

a. Kontak : gambar simbol kontak ini menggambarkan kondisi logika pada input yang dapat dianalogikan dengan sakelar togel, kondisi internal, sakelar tekan dsb.
b. Koil : gambar simbol koil ini mewakili output yang dianalogikan pada lampu, motor, starter, solenoid, relay, kondisi output internal dsb.

c. Kotak : gambar simbol kotak ini mewakili instruksi-instruksi tambahan seperti instruksi timer, counter atau instruksi matematika.

Contoh diagram ladder ditunjukkan pada gambar di bawah ini.



Gambar 20 Contoh Diagram Ladder

5. Instruksi Ladder Diagram

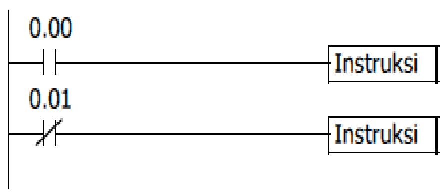
Instruksi diagram ladder adalah instruksi sisi kiri yang mengkondisikan instruksi lain di sisi kanan. Pada program diagram ladder instruksi ini disimbolkan dengan kontak-kontak seperti pada rangkaian kendali elektromagnet.

Instruksi diagram ladder terdiri atas enam instruksi ladder dan dua instruksi blok logika. Instruksi blok logika adalah instruksi yang digunakan untuk menghubungkan bagian yang lebih kompleks.

a. Instruksi **LOAD** dan **LOAD NOT**

Instruksi LOAD dan LOAD NOT menentukan kondisi eksekusi awal, oleh karena itu, dalam diagram ladder disambung ke bus bar sisi kiri. Tiap instruksi memerlukan satu baris kode mneumonik. Kata "instruksi" mewakili sembarang instruksi lain yang dapat saja instruksi sisi kanan yang akan dijelaskan kemudian.

Diagram Ladder



Mneumonik

Alamat	Instruksi	Operand
00000	LD	0.00
00001	Instruksi	
00002	LD NOT	0.01
00003	Instruksi	

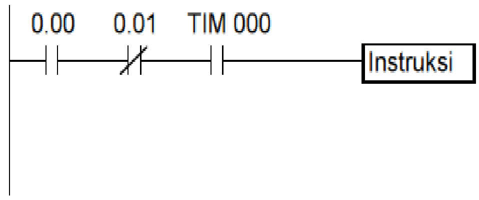
Gambar 21 Instruksi LOAD dan LOAD NOT

Jika misalnya hanya ada satu kontak seperti contoh di atas, kondisi eksekusi pada sisi kanan akan ON jika kontaknya ON. Untuk instruksi LD yang kontaknya NO, kondisi eksekusinya akan ON jika IR 0.00 ON; dan untuk instruksi LD NOT yang kontaknya NC, akan ON jika IR 0.01 OFF.

b. Instruksi **AND** dan **AND NOT**

Jika dua atau lebih kontak disambung seri pada garis yang sama, kontak pertama berkait dengan instruksi LOAD atau LOAD NOT dan sisanya adalah instruksi AND atau AND NOT. Contah di bawah ini menunjukkan tiga kontak yang masing-masing menunjukkan instruksi LOAD, AND NOT, dan AND.

Diagram Ladder



Mneumonik

Alamat	Instruksi	Operand
00000	LD	0.00
00001	AND NOT	0.01
00002	AND	TIM 000
00003	Instruksi	

Gambar 22 Instruksi AND dan AND NOT

c. Instruksi **OR** dan **OR NOT**

Jika dua atau lebih kontak terletak pada dua instruksi terpisah dan disambung paralel, kontak pertama mewakili instruksi LOAD atau LOAD NOT dan sisanya mewakili instruksi OR atau OR NOT. Contoh berikut menunjukkan tiga kontak yang masing-masing mewakili instruksi LOAD, OR NOT, dan OR.

Diagram Ladder



Mneumonik

Alamat	Instruksi	Operand
00000	LD	0.00
00001	OR NOT	0.01
00002	OR	TIM 000
00003	Instruksi	

Gambar 23 Instruksi OR dan OR NOT

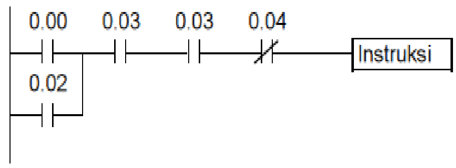
Instruksi akan mempunyai kondisi eksekusi ON jika salah satu di antara tiga kontak ON, yaitu saat IR 0.00 ON, saat IR 0.01 OFF, atau saat IR 0.03 ON.

d. Kombinasi Instruksi **AND** dan **OR**

Jika instruksi AND dan OR dikombinasikan pada diagram yang lebih rumit, mereka dapat dipandang secara individual di mana tiap instruksi menampilkan operasi logika pada kondisi eksekusi dan status bit operand. Perhatikan contoh

berikut ini hingga yakin bahwa kode mneumonik meliputi alur logika yang sama dengan diagram ladder.

Diagram Ladder



Mneumonik

Alamat	Instruksi	Operand
00000	LD	0.00
00001	AND	0.01
00002	OR	0.02
00003	AND	0.03
00004	AND NOT	0.04
00005	Instruksi	

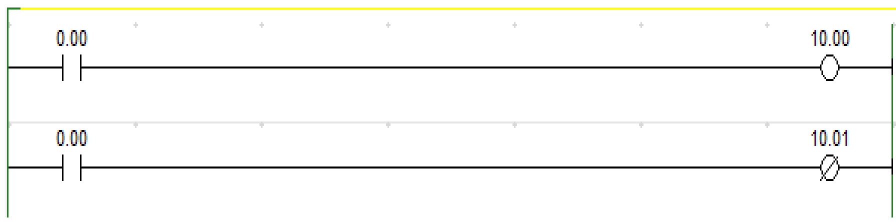
Gambar 24 Instruksi AND dan OR

Di sini AND terletak di antara statur IR 0.00 dan status IR 0.01 untuk menentukan kondisi eksekusi dengan meng-OR-kan status IR 0.02. Hasil operasi ini menentukan kondisi eksekusi dengan meng-AND-kan status IR 0.03 yang selanjutnya menentukan kondisi eksekusi dengan meng-AND-kan kebalikan status IR 0.04.

e. Instruksi **OUT** dan **OUT NOT**

Cara paling sederhana untuk meng-OUTPUT-kan kombinasi kondisi eksekusi adalah dengan meng-OUTPUT-kan langsung menggunakan instruksi OUTPUT dan OUTPUT NOT. Istruksi ini digunakan untuk mengendalikan status bit operand sesuai dengan kondisi eksekusi.

Dengan instruksi OUTPUT, bit operand akan ON selama kondisi eksekusinya ON dan akan OFF selama kondisi eksekusinya OFF. Dengan instruksi OUTPUT NOT, bit operand akan ON selama kondisi eksekusinya OFF dan akan OFF selama kondisi eksekusinya ON.



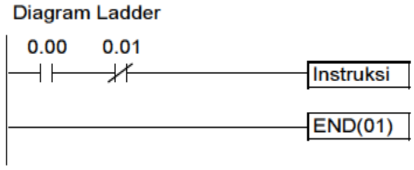
Gambar 25 Instruksi OUT dan OUT NOT

Pada contoh di atas, IR 10.00 akan ON jika IR 0.00 ON dan IR 10.01 akan OFF selama IR 0.01 ON. Di sini IR 0.00 dan IR 0.01 merupakan bit input dan IR 10.00 dan IR 10.01 merupakan bit output yang ditetapkan untuk peralatan yang dikendalikan PLC.

f. Instruksi **END(01)**

Instruksi terakihir yang diperlukan untuk melengkapi suatu program adalah instruksi END. Saat PLC menscan program, ia mengeksekusi semua instruksi hingga instruksi END pertama sebelum kembali ke awal program dan memulai eksekusi lagi. Meskipun instruksi END dapat ditempatkan sembarang titik dalam program, tetapi intruksi setelah instruksi END pertama tidak akan diekseksekusi.

Diagram Ladder



Mneumonik

Alamat	Instruksi	Operand
00000	LD	0.00
00001	AND NOT	0.01
00002	Instruksi	
00003	END(01)	

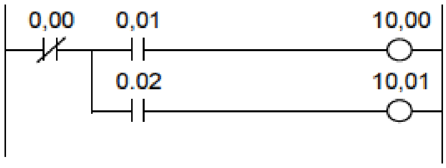
Gambar 26 Instruksi END (01)

Nomor yang mengikuti instruksi END dalam kode mneumonik adalah kode fungsinya, yang digunakan saat memasukkan instruksi ke dalam PLC menggunakan konsol pemrogram.

Instruksi END tidak memerlukan operand dan tidak boleh ada kontak ditempatkan pada garis instruksi yang sama. Jika dalam program tidak ada instruksi END, program tersebut tidak akan dieksekusi.

g. Penggunaan **Bit TR**

Bit TR (Temporarily Relay) digunakan untuk mempertahankan kondisi eksekusi pada garis instruksi bercabang. Hal ini dipertahankan karena garis instruksi dieksekusi menuju ke instruksi sisi kanan sebelum kembali ke titik cabang untuk mengeksekusi instruksi lainnya. Jika ada kontak pada garis instruksi setelah titik cabang, kondisi eksekusi untuk instruksi yang pertama tidak sama dengan kondisi pada titik cabang sehingga untuk mengeksekusi instruksi berikutnya menggunakan kondisi eksekusi titik cabang dan kontak lain setelah titik cabang tersebut.



Alamat	Instruksi	Operand
00000	LD NOT	0,00
00001	OUT	TR0
00002	AND	0.01
00003	OUT	10.00
00004	LD NOT	TR0
00005	AND	0.02
00006	OUT	10.01

Gambar 27 Penggunaan Bit TR.

Jika program dibuat dalam bentuk diagram ladder, tidak perlu memperhatikan bit TR karena bit TR hanya relevan pada pemrograman bentuk mneumonik.

Terdapat delapan bit TR, yaitu TR0 sampai dengan TR7 yang dapat digunakan untuk mempertahankan kondisi eksekusi sementara.

Misalkan suatu bit TR ditempatkan pada suatu titik cabang, kondisi eksekusinya akan disimpan pada bit TR tersebut. Jika kembali ke titik cabang, bit TR mengembalikan kondisi eksekusi yang telah disimpan.

Penyimpanan kondisi eksekusi pada titik cabang menggunakan bit TR sebagai operand dari instruksi OUTPUT. Kondisi eksekusi ini kemudian dikembalikan setelah mengeksekusi instruksi sisi kanan dengan menggunakan bit TR yang sama sebagai operand dari instruksi LOAD.

Contoh berikut ini menunjukkan penggunaan dua bit TR yaitu TR0 dan TR1 pada sebuah program.

h. Penggunaan Bit Kerja (Internal Relay)

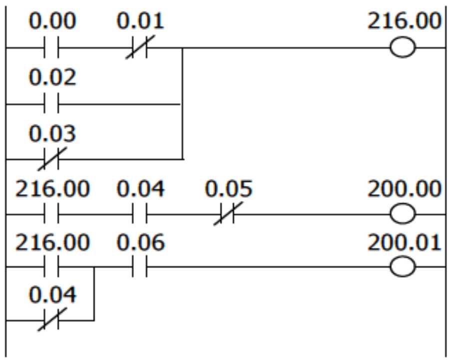
Dalam pemrograman, mengkombinasikan kondisi untuk menghasilkan kondisi eksekusi secara langsung sering sangat sulit. Kesulitan ini dapat siatasi dengan mudah menggunakan bit kerja untuk mentriger instruksi lain secara tidak langsung.

Bit kerja tidak ditransfer dari atau ke dalam PLC. Semua bit pada daerah IR yang tidak dialokasikan sebagai bit input/output dan bit pada daerah AR (Auxilary Relay) dapa digunakan sebagai bit kerja. Bit input/output dan bit yang dialokasikan untuk keperluan tertentu tidak dapat digunakan sebagai bit kerja.

Jika mengalami kesulitan pada pemrograman suatu program pengendalian pertimbangan pertama harus diberikan pada bit kerja untuk menyederhanakan program.

Bit kerja sering digunakan sebagai operand untuk salah satu instruksi OUTPUT, OUTPUT NOT, DIFERENTIATE UP, DIFERENTIATE DOWN, dan KEEP, kemudian digunakan sebagai

kondisi yang menentukan bagaimana instruksi lain dieksekusi. Bit kerja juga dapat digunakan untuk menyederhanakan program saat kombinasi kondisi tertentu digunakan berulang-ulang. Pada contoh berikut ini IR 0.00, IR 0.01, IR 0.02, dan IR 0.03 dikombinasikan pada blok logika yang menyimpan kondisi eksekusinya sebagai status IR 216.00. Kemudian IR 216.00 dikombinasikan dengan kontak lain untuk menentukan kondisi output untuk IR 200.00 dan IR 200.01.



Alamat	Instruksi	Operand
00000	LD	0.00
00001	AND NOT	0.01
00002	OR	0.02
00003	OR NOT	0.03
00004	OUT	216.00
00005	LD	216.00
00006	AND	0.04
00007	AND NOT	0.05
00008	OUT	200.00
00009	LD	216.00
00010	OR NOT	0.04
00011	AND	0.06
00012	OUT	200.01

Gambar 28 Penggunaan Bit Kerja.

i. **Timer** dan **Counter**

Pada sebagian besar aplikasi kontrol terdapat peralatan untuk beberapa aspek kontrol pewaktuan (timing). PLC mempunyai fasilitas pewaktuan untuk program yang dapat digunakan. Metode umum dari pemrograman sebuah rangkaian timer adalah untuk menentukan interval pewaktuan yang dihitung dari suatu kondisi atau keadaan dikenal dengan istilah timer.

Sedangkan fasilitas menghitung suatu kejadian (*event)* utuk menghitung banyaknya kejadian disebut counter. Counter digunakan untuk menghitung input yang masuk ke dalam counter tsb. No Counter = 0 – 255 ,No Timer = 0 – 255 Set Timer = #0000 - #9999s Perlu di ingat bahwa dalam membuat

program alamat/penomeran Counter dan Timer tidak boleh sama, misalnya anda membuat program memakai 3 counter dan 3 timer anda bisa pakai no. counter 0 – 2 sedangkan no. timernya anda pakai 4 - 6 dan seterusnya tergantung kebutuhan.

j. Instruksi **Timer**

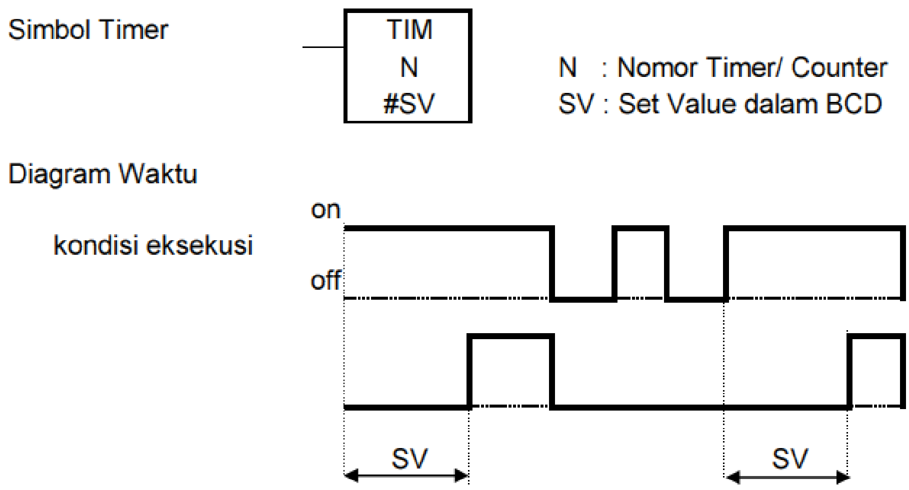
Timer adalah sebuah instruksi menunggu dalam waktu tertentu untuk melakukan sesuatu. Jika kita lihat timer dapat dimanfaatkan untuk berbagai keperluan dan untuk tipe timer diproduksi oleh pabrik disesuaikan dengan metoda dan cara mereka masing-masing.

Instruksi Timer digunakan untuk operasi tunda waktu. Ia memerlukan dua operand yang terletak pada dua baris instruksi, yaitu baris pertama untuk nomor timer dan yug kedua untuk settig waktu (SV = Set Value). Meskipun demikian, instruksi Timer terletak dalam satu alamat.

Nomor Timer dipakai bersama untuk nomor Counter. Nomor Timer/ Counter hanya boleh digunakan sekali. Maksudnya, sekali nomor Timer/ Counter telah digunakan, ia tidak boleh digunakan untuk instruksi Timer/ Counter yang lain. Tetapi, nomor timer sebagai operand suatu kontak dapat digunakan sebanyak yang diperlukan.

Banyaknya nomor Timer/ Counter bergantung kepada tipe PLC. Misalnya, PLC OMRON CPM1A, terdapat 128 nomor, yaitu dari 000 sampai dengan 127. tidak diperlukan awalan apapun untuk menyatakan nomor timer. Tetapi, jika nomor timer sebagai operand suatu kontak harus diberi awalan TIM. SV dapat berupa konstanta atau alamat channel/ words. Jika channel

daerah IR sebagai unit input dimasukkan sebagai alamat channel, unit input ini harus disambung sedemikian sehingga SV dapat diset dari luar. Timer/ Counter yang disambung dengan cara ini hanya dapat diset dari luar dalam mode MONITOR atau RUN. Semua SV, termasuk yang diset dari luar harus dalam BCD (Binary Coded Decimal), yaitu bilangan desimal yang dikode biner. Penulisan SV harus diawali dengan tanda #.



Gambar 29 Diagram waktu Instruksi Timer.

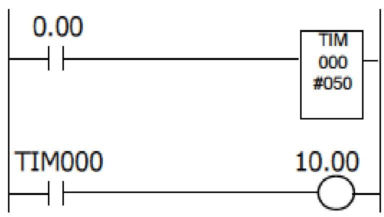
Timer bekerja saat kondisi eksekusinya beralih ke on dan direset (ke SV) saat kondisi eksekusinya beralih ke off. Jika kondisi eksekusi lebh lama daripada SV, *completion flag,* yaitu tanda yang menunjukkan hitungan waktu telah berakhir, tetap on hingga Timer direset. Timer akan reset jika trletak pada bagian program interlock saat kondisi eksekusi instruksi interlock (IL) off, dan saat terjadi pemutusan daya. Jika dikehendaki timer tidak reset oleh dua keadaan tersebut, maka bit pulsa clock pada daerah SR untuk mencacah Counter yang menghasilkan Timer menggunakan instruksi Counter. SV

mempunyai harga antara 0000 sampai dengan 9999 (BCD) dalam satuan deci-detik. Jadi, misalnya menghendaki 10 detik, maka nilai SV harus 100. Jika SV dinyatakan tidak dalam BCD, akan muncul pesan kesalahan.

Di bawah ini diberikan program-program penerapan timer.

1) Tunda on (1)

Jika kondisi eksekusi timer (hanya ditentukan oleh kontak 0.00) on, maka timer aktif. Lima detik kemudian (completion flag timer on) kontak TIM 000 on hingga selanjutnya output 10.00 on. Jika lama kontak 0.00 on lebih pendek daripada SV, maka completion flag tetap off dan output 10.00 juga tetap off.

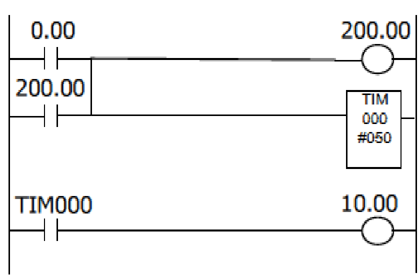


Alamat	Instruksi	Operand
00000	LD	0.00
00001	TIM	0.00
		#050
00002	LD	TIM 000
00003	OUT	10.00

Gambar 30 Program tunda ON (1).

Agar dapat aktif meskipun kontak 0.00 hanya on sesaat, gunakan bit kerja untuk mengendalikan timer secara tidak langsung seperti ditunjukkan pada program berikut ini:

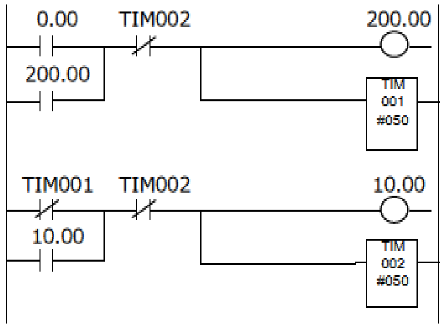
2) Tunda on (2)



Alamat	Instruksi	Operand
00000	LD	0.00
00001	OR	200.00
00002	OUT	200.00
00003	TIM	000
		#050
00004	LD	TIM 000
00005	OUT	10.00

Gambar 31 Program tunda ON (2).

3) Tunda ON dan OFF (3)



Alamat	Instruksi	Operand
00000	LD	0.00
00001	OR	200.00
00002	AND NOT	TIM 002
00003	OUT	200.00
00004	TIM	001
		#050
00005	LD NOT	TIM 001
00006	OR	10.00
00007	AND NOT	TIM 002
00008	OUT	10.00
00009	TIM	002
		#050

Gambar 32 Program tunda ON dan OFF.

k. Instruksi **Counter**

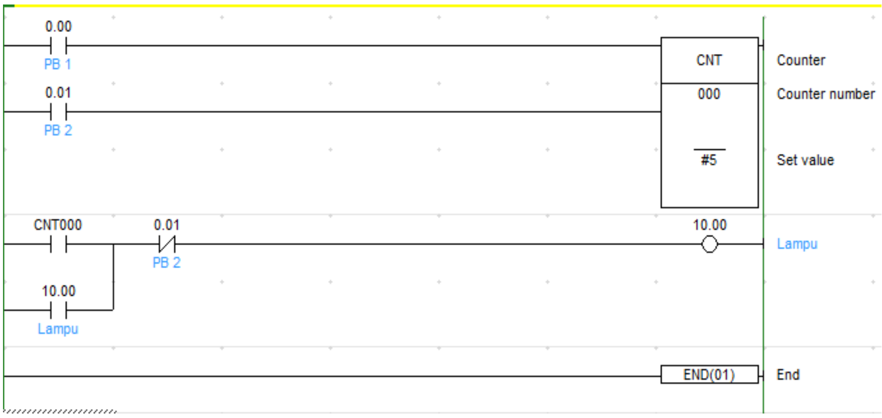
Penghitung/pencacah merupakan instruksi sederhana untuk melaksanakan hitungan dari setiap kejadian tergantungan data yang masuk pada input PLC. Alamat yang digunakan counter sama dengan alamat timer.

Nilai Timer/Counter pada PLC OMRON bersifat menghitung mundur dari nilai awal yang ditetapkan oleh program, setelah mencapai angka nol maka contact NO timer/counter akan ON.

Instruksi **CNT** berfungsi sebagai penghitung/pencacah mundur. **Apa yang dihitung?** Yang dihitung adalah perubahan kondisi masukan **CP** (Count Pulse) dari OFF ke ON.Ketika kondisi eksekusinya ON, maka setiap kali ada perubahan kondisi masukan CP dari ON ke OFF, maka instruksi CNT akan mengurangi nilai **PV-**nya (Present Value) dengan satu. Perubahan CP selain dari kondisi OFF ke ON tidak berpengaruh terhadap nilai PV.Jika PV telah mencapai nol, maka **Completion Flag** Counter akan ON. Kondisi tersebut akan dipertahankan sampai Counter direset.

Counter direset melalui kaki masukan **R.** Jika kondisi R berubah dari OFF ke ON, maka nilai PV akan direset menjadi sama dengan **SV.** Pada saat Counter dalam kondisi direset (R=ON), perubahan kondisi pada CP tidak akan berpengaruh pada PV.

Seperti halnya Timer instruksi CNT memiliki 2 operand yakni **TC Number** dan **SV** (Setting Value). TC Number dapat bernilai 0-255 untuk CPM2A dan 0-127 untuk CPM1A. Sedangkan SV dapat berupa konstanta (BCD) atau salah satu dari register IR, AR, SR, HR, LR, dan DM.



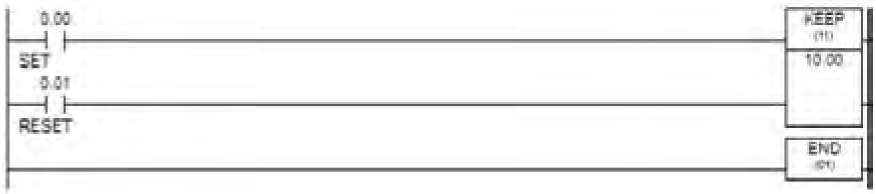
Gambar 33 Program Counter.

Counter umumnya dapat menghitung dari 0 sampai 9999, dari –32.768 sampai +32.767 atau dari 0 sampai 65535. Mengapa demikian?, karena PLC mempunyai penghitung 16 bit, dan ditentukan bahwa untuk penghitung 0-9999 adalah 16 bit penghitung BCD (binary coded decimal) dan yang menhitung dari -32,768 - 32767 dan 0 to 65535 - adalah 16-bit penghitung biner.

I. Instruksi Lanjut

1) Keep (11) Latching Relay

Keep digunakan seperti latch. Fungsi ini akan mempertahankan status bit ON atau OFF sampai ada satu dari dua input yang menge-set atau reset fungsi tersebut. Bila fungsi KEEP ini digunakan dengan HR relay,instruksi dari output latch akan dipertahankan selama terjadi gangguan daya. Instruksi ini digunakan untuk mempertahankan kondisi bit operand berdasarkan dua kondisi eksekusi, yakni Set dan Reset.



Gambar 34 Program KEEP.

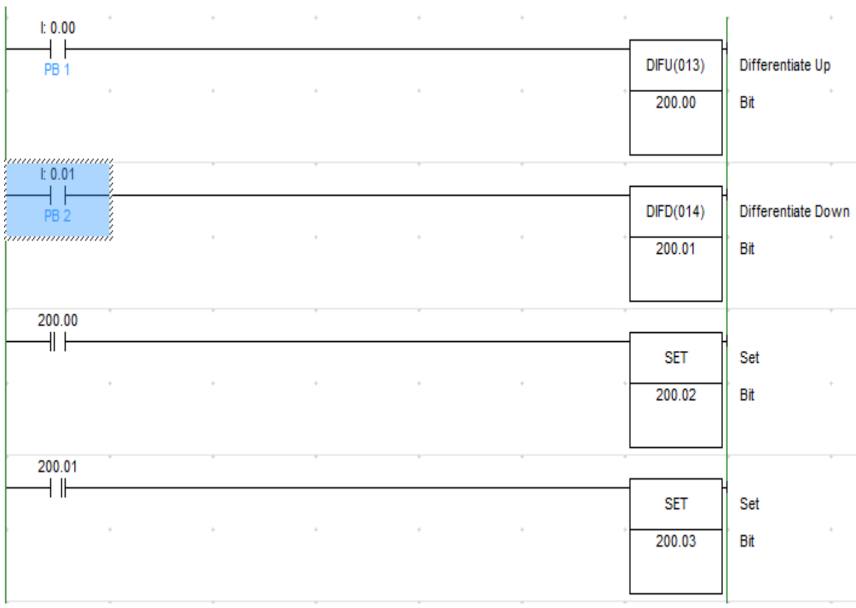
Oleh sebab itu, instruksi KEEP ini terhubung ke dua baris instruksi pengkondisi eksekusi. Jika kondisi eksekusi instruksi pada baris pertama ON (Set), maka kondisi bit operand instruksi KEEP akan ON. Dan jika kondisi eksekusi instruksi pada baris kedua ON (Reset), maka kondisi bit operand instruksi KEEP akan OFF. Jadi instruksi KEEP ini seperti instruksi SET dan RSET yang dijadikan satu paket.

2) Instruksi Differensial DIFU dan DIFD

Instruksi DIFU(13) dan DIFD (14) output akan menjadi ON hanya dalam satu waktu scan.

DIFU output akan ON saat terjadi transisi OFF ke ON pada sinyal inputnya.

DIFD output akan menjadi ON saat terjadi transisi ON ke OFF pada sinyal inputnya.



Gambar 35 Program DIFU dan DIFD

3) Instruksi JMP-JME

Instruksi Jump – JMP(04) selalu berpasangan dengan instruksi Jump End – JME(05). JMP digunakan untuk melewati bagian program tertentu dalam program, yakni bagian program yang terletak di antara instruksi JMP(04) dan JME(05). Kita sebut saja bagian program tersebut sebagai **jump section.**

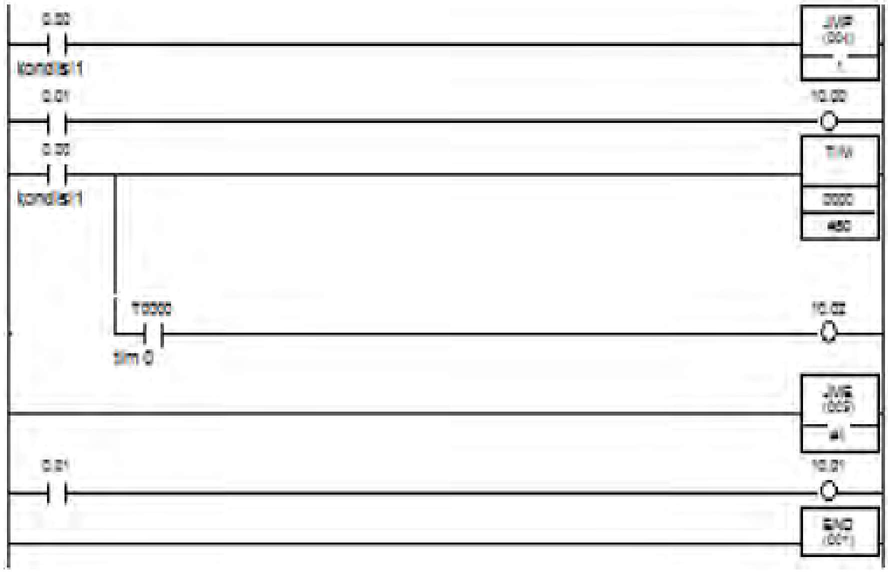
Jika kondisi eksekusi JMP(04) ON, maka program akan berjalan „lurus" seperti jika tidak ada instruksi JMP(04). Dengan kata lain, instruksiinstruksi dalam jump section akan dieksekusi. Tapi jika kondisi eksekusi JMP(04) OFF, maka eksekusi program akan melompat menuju instruksi tepat di bawah instruksi JME(05), **tanpa mengubah** nilai status apapun yang ada di dalam **jump section.**

Terdapat dua tipe Instruksi JMP(04) dan JME(05), yakni instruksi dengan nomor 01-99, dan instruksi dengan nomor 00. Instruksi dengan nomor 01-99 hanya dapat digunakan

satu kali dalam program. Pada kondisi eksekusi OFF, instruksi JMP(04) dengan nomor 01 (JMP(04)@01) akan melompatkan eksekusi program ke JME(05) dengan nomor 01 (JME(05)@01), seolah-olah instruksi-instruksi yang ada di antara keduanya tidak ada.

Instruksi JMP(04)@00 dapat digunakan berulang-ulang dalam program.

Bahkan bisa juga instruksi ini digunakan secara berurutan dengan hanya satu instruksi JME(04) saja. Dalam eksekusinya, program akan mencari pasangan JMP(04)@00 terdekat pada instruksi-instruksi selanjutnya meskipun tanpa mengubah nilai status apapun. Akan tetapi hal tersebut menyebabkan proses eksekusi program menjadi sedikit lebih lama dibandingkan ketika menggunakan instruksi JMP(04) dan JME(05) dengan nomor 01-99.



Gambar 36 Program JMP dan JME

Pada saat **Kondisi1** OFF, eksekusi program akan langsung melompat ke instruksi tepat di bawah JME(05). Perubahan

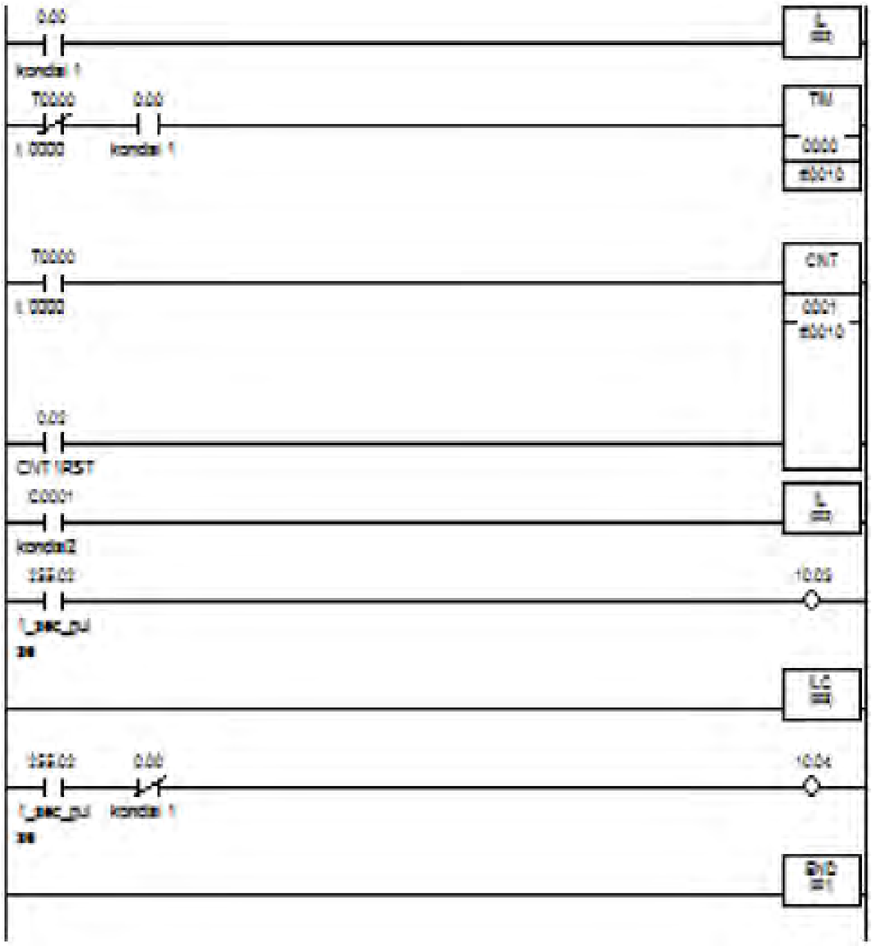
nilai **Input 000.01** hanya akan mengubah kondisi **Output 010.01** saja, sedangkan kondisi **Output 010.00** tetap.

Jika **Kondisi1** ON, maka program akan mengeksekusi **0.01 (10.00)** hingga **0.01 (10.01). Timer0** akan bekerja, dan perubahan nilai pada **Input 000.01** akan mengubah kondisi **Output 010.00** dan **Output 010.01.**

4) Instruksi IL-ILC

Interlock – IL(02) digunakan bersama dengan instruksi Interlock Clear – ILC(03). Instruksi ini digunakan untuk menjalankan bagian program tertentu, yakni bagian program yang berada di antara IL(02) dan ILC(03), dengan syarat kondisi eksekusi untuk IL(02) terpenuhi (ON). Bagian program yang terletak di antara IL(02) dan ILC(03) disebut sebagai **interlock section.**

IL(02) dan ILC(03) tidak harus berpasangan satu-satu. IL(02) dapat digunakan berkali-kali dengan satu ILC(03) penutup. Setiap IL(02) membentuk bagian interlock-nya masing-masing hingga bertemu dengan ILC(03). Akan tetapi tidak memungkinkan untuk membuat interlock bersarang (*nested interlock).*



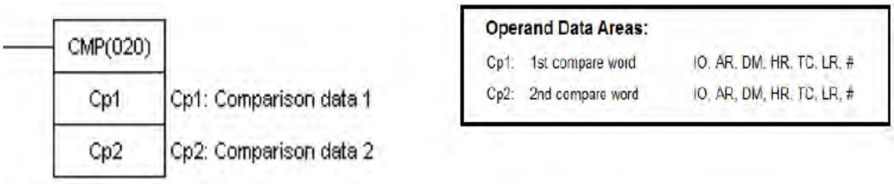
Gambar 37 Program IL dan ILC

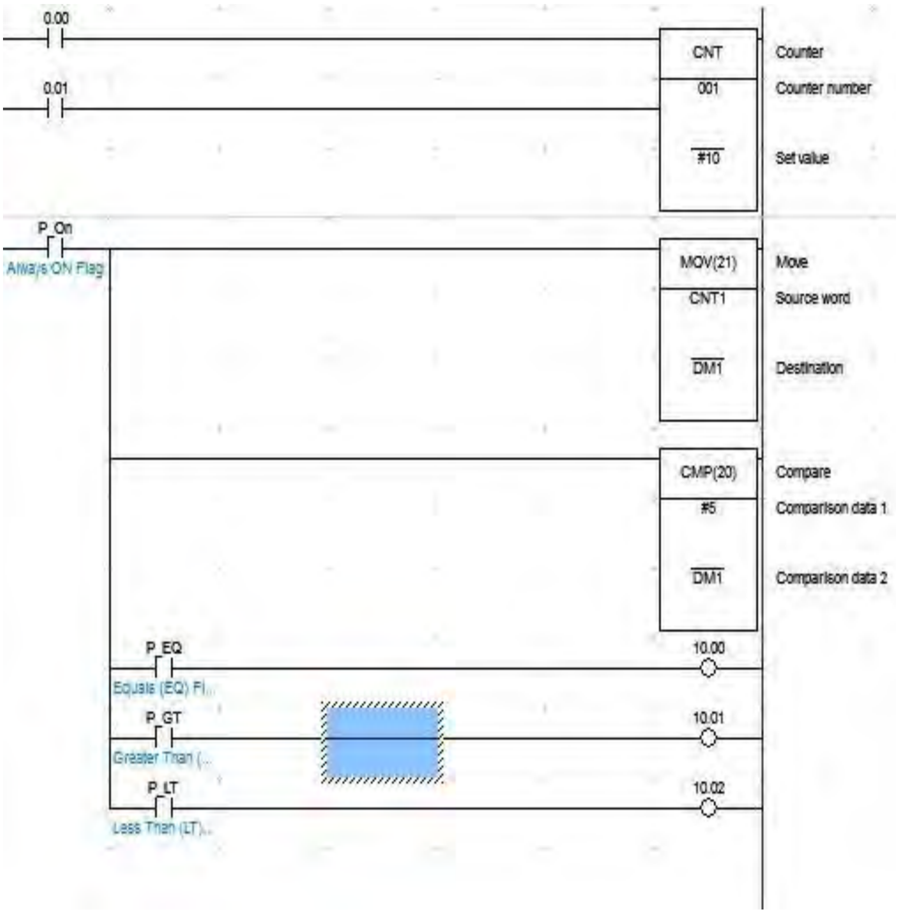
5) Instruksi Compare – CMP (20)

Compare (CMP) digunakan untuk membandingkan data dalam kanal tertentu dengan data kanal yang lain,atau sebuah empat digit,konstanta heksadesimal. Instruksi **CMP(20)** berfungsi membandingkan dua buah operand bertipe word. Ketika kondisi eksekusi instruksi ini terpenuhi, maka CMP(20) akan membandingkan nilai **operand1** dengan nilai **operand2.** Hasil perbandingan tersebut disimpan dalam bit flag **EQ** (EQuals), **LE** (LEss-than), dan **GR** (GReater-than) yang menyatakan **operand1 =**

operand2, operand1 < operand2, dan **operand1 > operand2.**

Simbol Ladder





Gambar 38 Program Compare

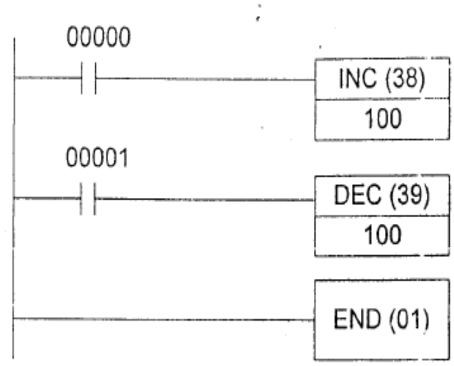
6) Instruksi Increment dan Decrement

Increment adalah kemampuan Instruksi ini berfungsi untuk menambah satu nilai pada *operand* bertipe *word. Operand*

dalam hal ini bisa salah satu dari register IR, SR, AR, DM, HR, dan LR. Instruksi ini umumnya disebut instruksi INC.

Decrement adalah kemampuan untuk mengurangi satu nilai pada *operand* bertipe *word. Operand* dalam hal ini bisa salah satu dari register IR, SR, AR, DM, HR, dan LR. Instruksi ini umumnya disebut instruksi DEC.

Diagram Ladder



Kode Mnemonik

Alamat	Instruksi	Data
00000	LD	00000
00001	INC (38)	
		100
00002	LD	00001
00003	DEC (39)	
		100
00004	END (01)	

Gambar 39 Program INC dan DEC

7) Instruksi Aritmatika

Instruksi Aritmatika adalah fungsi matematika di dalam PLC, sering kali dalam aplikasi kita harus menggunakan rumus matematika pada data kita. Hal ini jarang terjadi akan tetapi pada kondisi tertentu data tersebut *betul-betul kita butuhkan.* Pada umumnya hampir semua PLC mengikut sertakan fungsi matematika berikut:

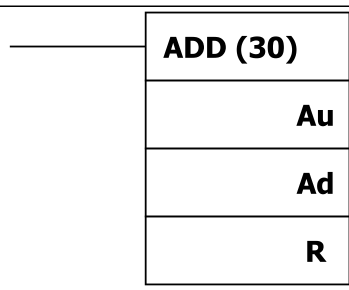
Instruksi Aritmatika ini digunakan untuk melaksanakan fungsi matematis seperti penambahan, pengurangan, pembagian, dan perkalian. Tipe data yang dapat digunakan pada instruksi matematis ini adalah : Integer, Double Integer, Real Instruksi Increment dan Decrement merupakan instruksi yang digunakan untuk penambahan dan pengurangan secara bertahap pada suatu data Byte, Word atau Doube Word.

Penjumlahan	**:**	Kemampuan menambahkan satu data ke data lainnya, instruksi ini umumnya disebut instruksi ADD.
Pengurangan	**:**	Kemampuan mengurangkan satu data dengan data lainnya, instruksi ini umumnya disebut instruksi SUB.
Multiplikasi	**:**	Kemampuan mengalikan satu data dengan data lainnya, instruksi ini umumnya disebut instruksi MUL.
Pembagian	**:**	Kemampuan membagi satu data dengan data lainnya, instruksi ini umumnya disebut instruksi DIV.

a. Add – ADD (30)

Simbol pada Ladder Area data

Au : Augend word (BCD)
IR, SR, AR, DM, HR, TC, LR, #
Ad : Addend word (BCD)
IR, SR, AR, DM, HR, TC, LR, #
R : Result word
IR AR DM HR LR



Gambar 40 Simbol dan Area data ADD

ADD menjumlahkan data dari dua *channel* yang berbeda, atau satu channel dan satu konstanta yang akan

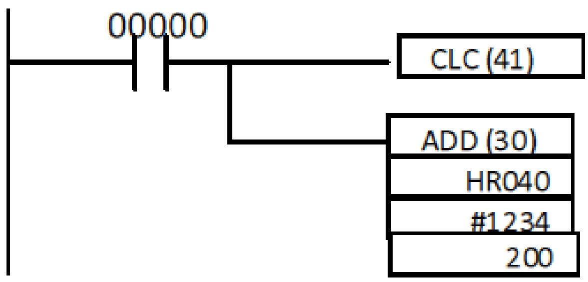
memberikan output pada *channel* yang ke tiga. Karena itu, tiga parameter data harus ditentukan : *augend* (penjumlah), *addend* (yang dijumlahkan), dan *result* (hasil).

Operasi yang terjadi pada instruksi ADD ialah sebagai berikut :

Au + Ad + CY → CY R

→ dimana CY ialah *carry flag*

Diagram Ladder



Gambar 41 Program aritmatik penjumlahan

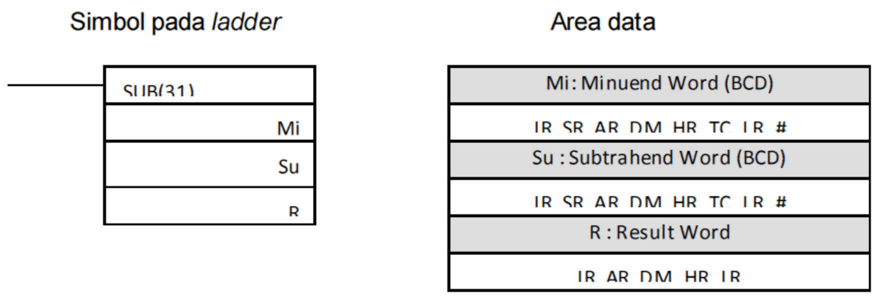
Dari program di atas, ketika input 00000 dinyalakan, data pada *internal relay* **HR040** dijumlahkan dengan konstanta **1234.** Hasilnya ditampilkan ke **CH 200.**

Jika *carry* dihasilkan akibat penjumlahan, *carry flag* (SR 25504) akan **ON.**

Dalam contoh di atas, sebelum mengeksekusi ADD, *carry flag*/CY (*special relay* 25504) akan dimatikan oleh *Clear Carry* (CLC).

Augend and *addend* harus dalam bentuk **BCD** (0 sampai 9999), jika tidak *special relay* 25503 (*error flag)* akan ON dan instruksi ADD tidak akan dieksekusi.

b. Subtract – SUB (31)



Gambar 42 Simbol dan Area data SUB.

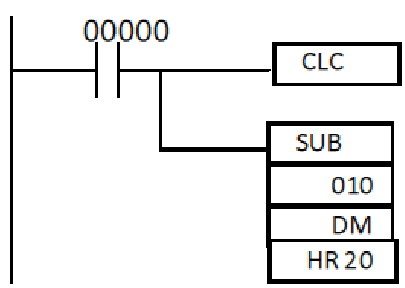
SUB mengurangkan data di **Mi** dengan data di **Su** dan **CY** (carry flag - 25504), dan meletakkan hasilnya di **R.**

Jika hasilnya negatif, **CY** akan aktif (ON) dan nilai *10's complement* dari hasil sebenarnya akan diletakkan di **R.** Untuk mendapatkan hasil sebenarnya, kurangkan 0 dengan hasil awal yang ada di **R.**

Operasi yang terjadi pada instruksi SUB ialah sebagai berikut :

$$Mi - Su - CY \rightarrow CY \quad R$$

dimana CY ialah *carry flag*



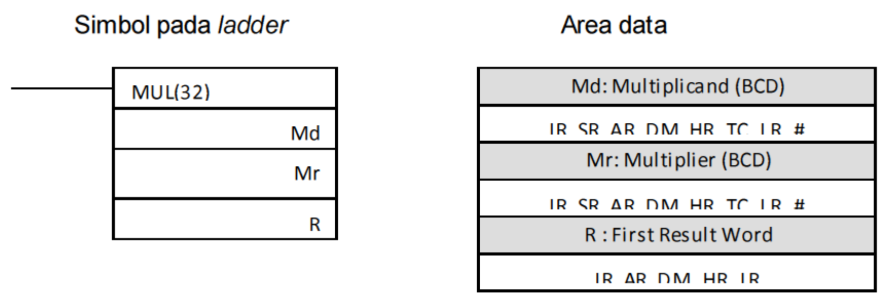
Gambar 43 Program aritmatik pengurangan

Dari program di atas, ketika input 00000 dinyalakan, data pada **IR 010** dikurangkan dengan data pada **DM 0100.** Hasilnya ditampilkan ke **HR 20.**

Dalam contoh di atas, sebelum mengeksekusi SUB, *carry flag/*CY (*special relay* 25504) akan dimatikan oleh *Clear Carry* (CLC).

Minuend dan *subtrahend* harus dalam bentuk **BCD** (0 sampai 9999), jika tidak *special relay* 25503 (*error flag)* akan ON dan instruksi SUB tidak akan dieksekusi.

c. Multiply – MUL (32)



Gambar 44 Simbol dan Area data MUL.

MUL mengalikan data di **Md** dengan data di **Mr,** dan meletakkan hasilnya di **R** dan **R+1 (R** dan **R+1** harus berada di area data yang sama).

Operasi yang terjadi pada instruksi MUL ialah sebagai berikut :

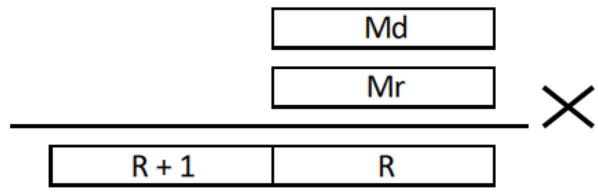
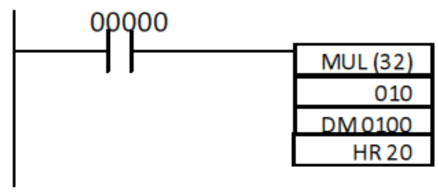


Diagram *ladder*

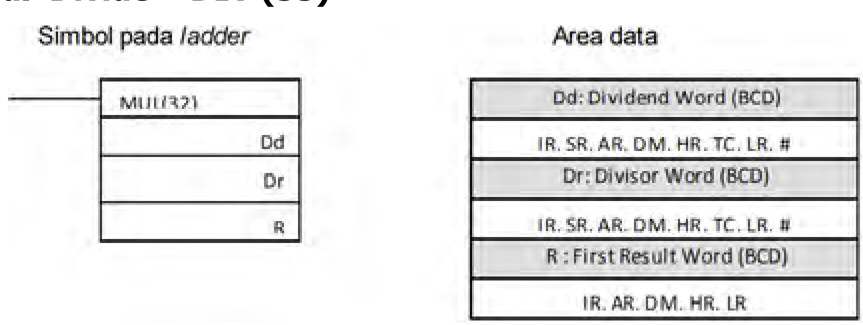


Gambar 45 Program aritmatik perkalian

Dari program di atas, ketika input 00000 dinyalakan, data di **IR 010** dikalikan dengan data di **DM 0100.** Hasilnya ditampilkan ke **HR 20** dan **HR 21.**

Multiplicand dan *multiplier* harus dalam bentuk **BCD** (0 sampai 9999), jika tidak *special relay* 25503 (*error flag)* akan ON dan instruksi MUL tidak akan dieksekusi.

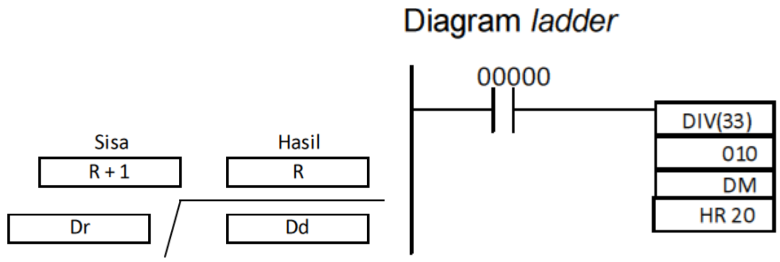
d. Divide – DIV (33)



Gambar 46 Simbol dan Area data DIV.

DIV membagi data di **Dd** dengan data di **Dr,** dan meletakkan hasilnya di **R** dan **R+1** **(R** dan **R+1** harus berada di area data yang sama). **R** berisi hasil pembagian, sedangkan **R+1** berisi sisa bilangan yang tidak habis dibagi.

Operasi yang terjadi pada instruksi DIV ialah sebagai berikut :



Gambar 47 Program aritmatik pembagian.

Dari program di atas, ketika 00000 dinyalakan, data di **IR 010** dibagi dengan data di **DM 0100.** Hasilnya ditampilkan di **HR 20** dan sisanya di **HR 21.**

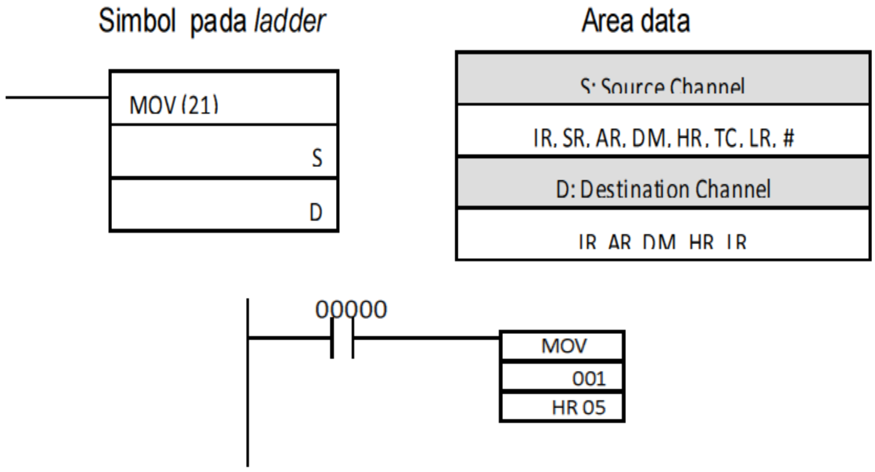
Dividend dan *divisor* harus dalam bentuk **BCD** (0 sampai 9999), jika tidak *special relay* 25503 (*error flag)* akan ON dan instruksi DIV tidak akan dieksekusi.

8) Instruksi Transfer Data

Instruksi **MOVE - MOV(21)** digunakan untuk meng-*copy* nilai dari **Source** ke **Destination. Source** dapat berupa konstanta (#), ataupun data yang ada di alamat tertentu dalam register IR, SR, AR, DM, HR, TC, dan LR. Sedangkan Destination adalah alamat register IR, SR, AR, DM, HR, LR. Jika kondisi eksekusi MOV(21) ON, maka data di **Source** (Sumber) akan di-*copy* ke **Destination** (Tujuan).

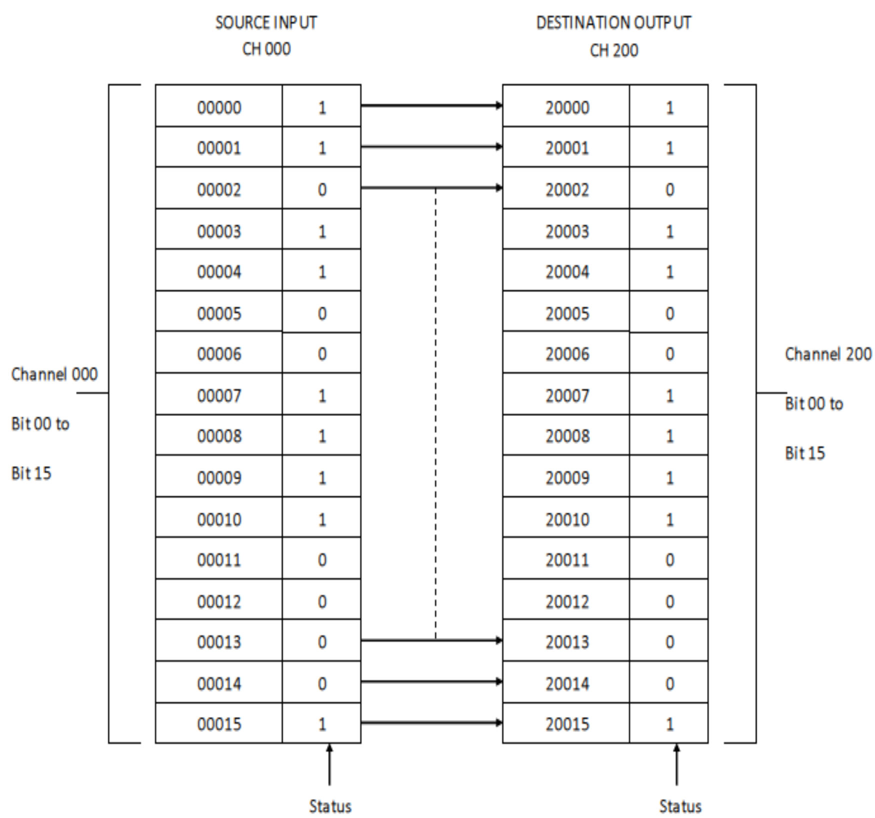
- Instruksi MOV(21) tidak dapat digunakan untuk mengubah nilai PV (Process Value) pada Timer/Counter.
- Instruksi MOV(21) tidak dapat digunakan untuk mengubah nilai DM6144 sampai DM6655.

- Instruksi MOV mentransfer data sumber (baik data dalam *channel* tertentu atau konstanta heksadesimal 4 digit) ke *channel* tujuan. Karena itu, MOV memerlukan dua parameter yang harus ditentukan : *channel* sumber atau konstanta dan *channel* tujuan



Gambar 48 Simbol dan Area data MOV dan diagram Ladder MOV.

Digram berikut menggambarkan operasi MOV :



Dalam kasus di atas, data pada Input Channel 000 dipindah ke Output Channel 200.

C. Tugas

Untuk memahami dan mendalami konsep pengoperasian PLC, maka kerjakan tugas dibawah ini !

1. Sebutkan cara-cara memprogram PLC !
2. Bagaimanakah langkah pembuatan program kontrol untuk PLC !
3. Untuk membuat *ladder diagram* diperlukan software, sebutkan software yang anda ketahui untuk memprogram PLC !

D. Tes Formatif

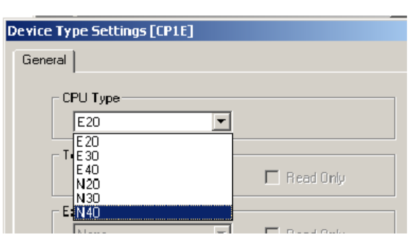
PETUNJUK KHUSUS :

Pilihlah salah satujawaban yang andaanggap paling benar !

6. Alamat output PLC omron CPM 2A yang dapat disambung dengan relay adalah :
   a. 10.16 c. 001 e. 10.00
   b. 10.20 d. 002

7. Sedangkan PLC omron CQM1H yang dapat disambung dengan relay adalah :
   a. 10.08 c. 001 e. 10.07
   b. 100.01 d. 002

8. Perhatikan gambar di bawah ini, untuk PLC CP1E I/O30, pilihan setting yang benar adalah:



a. N40
b. E20
c. N20
d. E30
e. N30

9. Icon dibawah ini merupakan perintah untuk :



a. Transfer program dari PLC
b. PLC Online

c. Download

d. PLC Run

e. Transfer program ke PLC

10. Contoh bahasa pemrograman PLC dengan instruksi adalah ….

a. Mnemonic code

b. Lader diagram

c. Logik diagram

d. Diagram I/O

e. Memori diagram

11. Instruksi "END" pada PLC terletak pada bagian :

a. Pertengahan Program

b. Input

c. Awal program

d. Akhir program

e. Output

12. Sebutkan enam macam instruksi diagram ladder dibawah ini...

a. LOAD, LOAD NOT, AND, AND NOT, OUT, OUT NOT

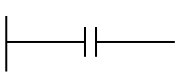
b. LOAD, LOAD NOT, AND, AND NOT, OUT, END

c. LOAD, LOAD NOT, AND, AND NOT, AND LOAD, OR LOAD

d. LOAD, LOAD NOT, AND, AND NOT, TIM, COUNT

e. LOAD, LOAD NOT, AND, AND NOT, OR, OR NOT

13. Gambar di bawah ini menunjukkan gambar:



a. Instruksi Load

b. Instruksi Load NOT

c. Instruksi AND

d. Instruksi AND NOT

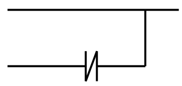
e. Instruksi OR

14. Gambar di bawah ini menunjukkan gambar:



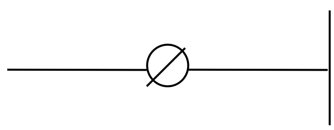
a. Instruksi Load
b. Instruksi Load NOT
c. Instruksi AND
d. Instruksi AND NOT
e. Instruksi OR

15. Gambar di bawah ini menunjukkan gambar:



a. instruksi load NOT
b. instruksi load OR NOT
c. instruksi load AND NOT
d. instruksi AND NOT
e. instruksi OR NOT

16. Gambar di bawah ini menunjukkan gambar:



a. instruksi out NOT
b. instruksi load OR NOT
c. instruksi load AND NOT
d. instruksi AND NOT
e. instruksi OR NOT

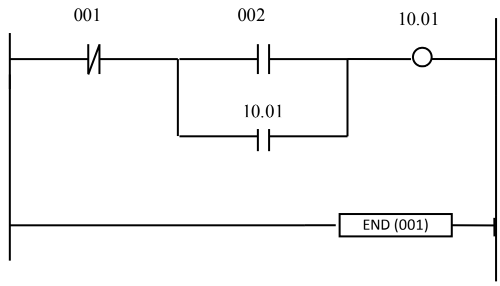
17. Sebutkan contoh instruksi yang tidak memerlukan operand!
   a. END(01), IL(02), ILC(03), JMP(04), MOV(21)
   b. AND, AND NOT, TIM, COUNT, END(01)
   c. END(01), IL(02), ILC(03), JMP(04), JME(05)
   d. OUT, OUT NOT, LOAD, LOAD NOT, IL(02)
   e. END(01), IL(02), ILC(03), JMP(04), TIM

18. Sebutkan contoh instruksi yang tidak memerlukan kondisi !

a. END(01) c. ILC(03) e. MOV(21)

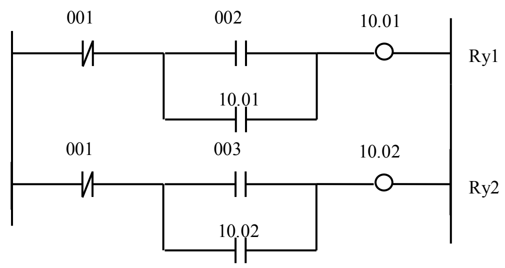
b. IL(02) d. JMP(04)

19. Cermati gambar di bawah ini, pernyataan yang salah dari gambardi bawah ini adalah:



a. merupakan rangkaian penguncian (laching)
b. rangkaian ini tidak bisa dimatikan
c. 001 untuk mematikan rangkaian terkunci
d. 002 untuk menghidupkan rangkaian
e. kontak 10.01sebagai pengunci

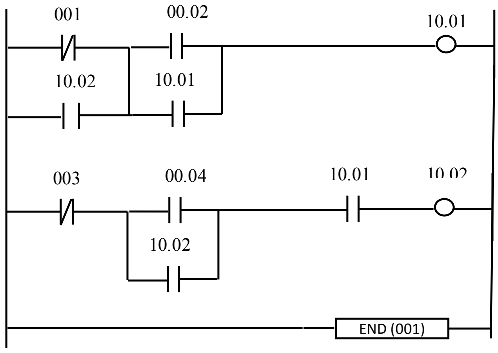
20. Cermati gambar di bawah ini, pernyataan yang benar dari ganbar di bawah ini adalah:



a. jika 002 ditekan Ry1 dan 2 On
b. merupakan rangkaian kerja berurutan
c. merupakan rangkaian kerja bergantian
d. jika 001 ditekan semua relay Off

e. kerja Ry1 tergantung dari kerja Ry2

21. Pernyataan yang salah dari gambar disamping adalah :



a. 004 berfungsi untuk mengaktifkan 10.02
b. jika 004 di on-kan maka 10.02 langsungaktif
c. untuk mematikan 10.02 dengan 003
d. kontak 10.02 untuk pengunci output 10.02
e. kontak 10.01 untuk pengunci output 10.01

22. Pada gambar soal nomor 23, jika dikehendaki output disambung dengan Relay luar, maka harus menggunakan PLC type :

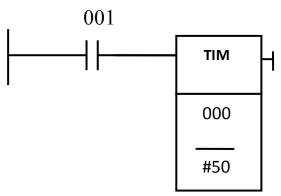
a. CP1E
c. CP1H
e. Semua type
b. CP1L
d. CPM2A

23. Dari gambar soal no 23 pada rangkaian tersebut, pernyataan yang salah adalah............

a. untuk menghidupkan harus urut 10.01 kemudian 10.02
b. rangkaian pengendali kerja bergantian
c. rangkaian pengendali kerja berurutan
d. untuk mematikan harus urut 10.02 kemudian 10.01

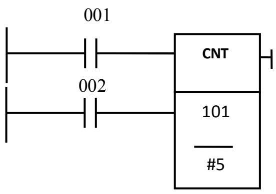
e. 10.01 tidak dapat dimatikan jika 10.02 masih aktif

24. Pernyataan yang salah dari gambar dibawah ini adalah :



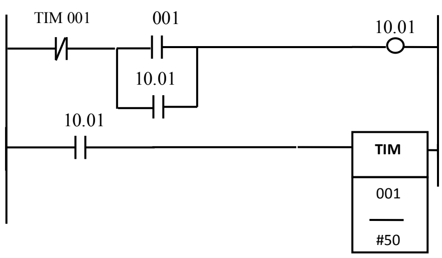
a. gambar timmer
b. diatur 50 detik
c. nomor timmer 0
d. 001 input timmer
e. 001 sebagai reset

25. Pernyataan yang salah dari gambar dibawah ini adalah :



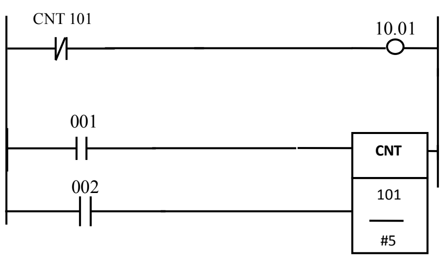
a. gambar counter
b. diatur 5 kali
c. nomor counter 1
d. 001 input counter
e. 002 sebagai reset

26. Dari gambar di bawah ini berlaku pernyataan-pernyataan:



a. rangkaian dihidupkan dari 001
b. rangkaian hidup selama 5 detik
c. rangkaian dimatikan oleh timmer
d. timmer bekerja setelah 50 detik
e. input timmer 10.01

27. Dari gambar di bawah ini berlaku pernyataan-pernyataan:



a. rangkaian dihidupkan dari 001
b. rangkaian hidup setelah 001 diaktifkan 5 kali
c. rangkaian dimatikan oleh counter

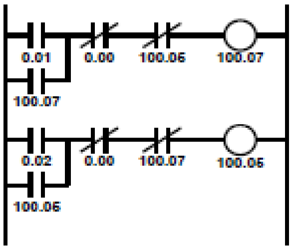
d. no counter adalah 01

e. 002 adalah reset counter

28. Pengguanaan Instruksi Timer digunakan untuk ......
    a. Operasi pengaktifan waktu
    b. Operasi tunda waktu
    c. Operasi penetapan waktu
    d. Operasi pengunduran waktu
    e. Operasi memutuskan daya

29. Apa yang dimaksud dengan SV (Set Value) ?
    a. Setelan untuk counter
    b. Setelan untuk Differential UP
    c. Setelan untuk Differential Down
    d. Setelan waktu untuk Timer
    e. Setelan untuk IL

30. Gambar Ladder Diagram dari pemprograman PLC dibawah ini digunakan untuk pengoperasian:



    a. Motor putar secara berurutan
    b. Motor putar secara interlock
    c. Motor putar secara otomatis
    d. Motor putar secara bintang-segitiga
    e. Motor putar secara bergantian / Forward Reverse

31. Langkah – langkah pembuatan program yang benar adalah.......

a. Membuat program kendali – menetapkan bit operand – menguraikan urutan kendali
b. Membuat operand – membuat program kendali – menguraikan urutan kendali
c. Mengguraikan urutan kendali – menetapkan bit operand – membuat program kendali
d. Menguraikan urutan kendali – membuat program kendali – membuat operand
e. Membuat operand – membuat program kendali – menguraikan urutan kendali

32. Jika suatu PLC OMRON terdiri 20 I/O dapat dikatakan dengan ……
a. Input dan Output sebanyak 20
b. Terminal Input dan Output 10
c. Input terdiri dari 8 dan Output terdiri dari 12
d. Input terdiri dari 12 dan Output terdiri dari 8
e. Input terdiri dari 10 dan Output terdiri dari 10

33. Tiap garis instruksi harus memiliki sedikitnya satu kondisi yang menentukan eksekusi instruksi sisi kanan, Kecuali ?

a. IL(02) c. END (01) e. TIM
b. JMP (04) d. MOV(21)

34. Garis intruksi yang diharuskan berada pada.....
a. terminal sisi kanan
b. Terminal pencabangan
c. Terminal akhir
d. terminal sisi kiri

e. Terminal tengah

35. Penetapan bit operand untuk peralatan I/O mengacu pada .....
   a. Daerah memori PLC
   b. Modul Interface I/O
   c. Internal Relay
   d. Daerah proses PLC
   e. Central Proscessing Unit (CPU)

E. Lembar Kerja

a) LANGKAH KERJA

Buat program PLC dengan perintah / instruksi yang sudah disampaiakn !

b) ALAT DAN BAHAN

- Trainer PLC
- PC Komputer / Laptop
- Kabel Jamper
- Obeng

c) KESELAMATAN DAN KESELAMATAN KERJA

- ✓ Gunakanlah pakaian praktik.
- ✓ Bacalah dengan seksama dan benar petunjuk praktikum.
- ✓ Hati-hati dengan aliran arus listrik.
- ✓ Jangan meletakkan peralatan di tepi meja.
- ✓ Kabel penghubung yang tidak terpakai jangan dekat dengan rangkaian.

- ✓ Tanyakan kepada instruktur hal-hal yang meragukan.

BAB 4

KEGIATAN BELAJAR 4 :
PEMBUATAN PROGRAM KONTROL PLC

A. Tujuan Pembelejaran

Setelah menyelesaikan kegiatan belajar 3, siswa diharapkan mampu :

1. Memahami dan mengaplikasikan penyusustnan program PLC.
2. Memahami dan mengaplikasikan instruksi dan simbol program PLC.
3. Memahami dan mengoperaskan software PLC.
4. Memahami dan mengaplikasikan rangkaian PLC pada rangkaian aplikasi kontrol.

B. Uraian Materi

Pemrograman PLC OMRON dengan CX Programmer

Pada dasarnya tahapan pembuatan program sampai program tersebut dapat dijalankan dapat dibagi ke dalam lima tahap sebagai berikut:

a) Persiapan

Langkah-langkah melakukan persiapan adalah sebagai berikut:

1) Pastikan komputer dan PLC Omron terkoneksi dengan baik.

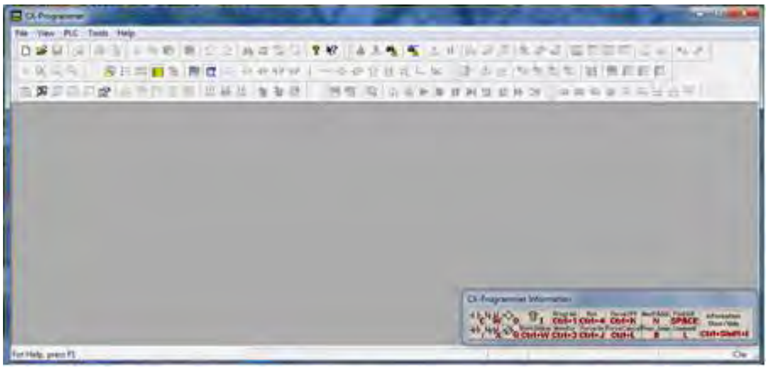
Hidupkan komputer dan PLC Unit.

2) Untuk membuka CX Programmer *double click* icon CX Programmer (Gambar 4.1) pada tampilan destop komputer. Dalam proses *loading* akan muncul tampilan seperti gambar 4.2 pada destop.



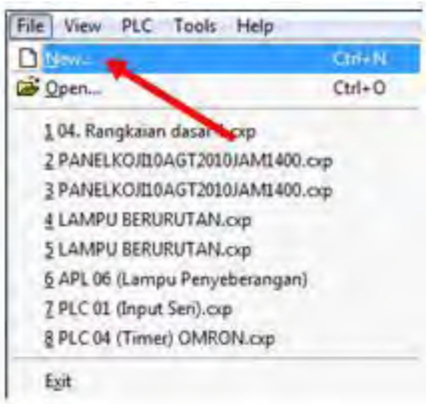
Gambar 49

3) Bila proses *loading* selesai, akan muncul tampilan seperti gambar 4.3



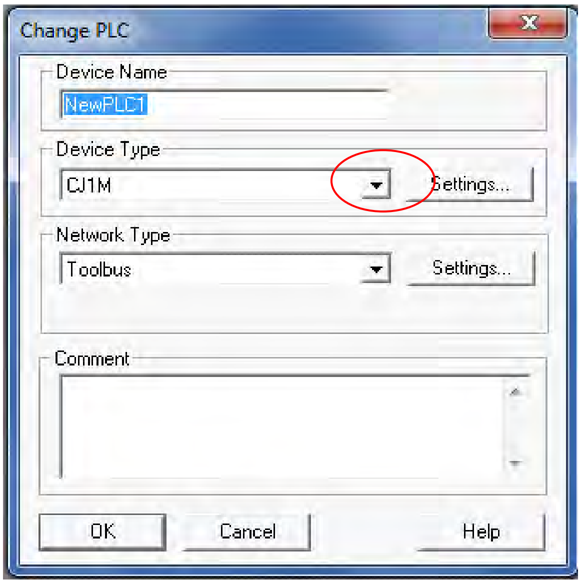
Gambar 50

4) Untuk memulai membuat program, klik ***File*** pada *menubar* kemudian pilih ***New***. (Perhatikan tanda panah pada Gambar 4.4)



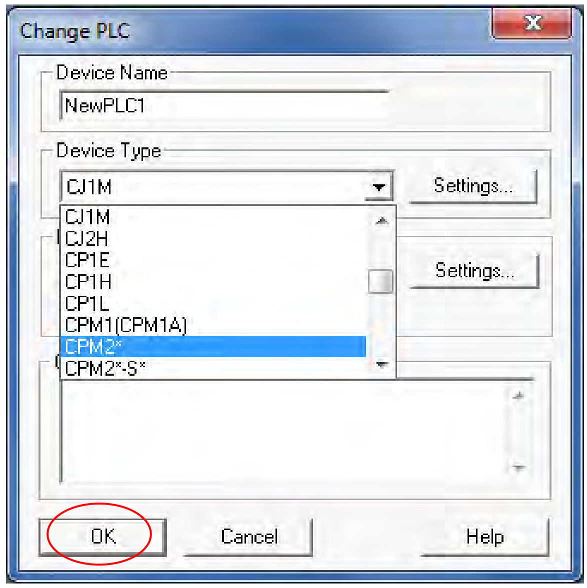
Gambar 51

5) Setelah diklik *New*, muncul tampilan seperti gambar 4.5



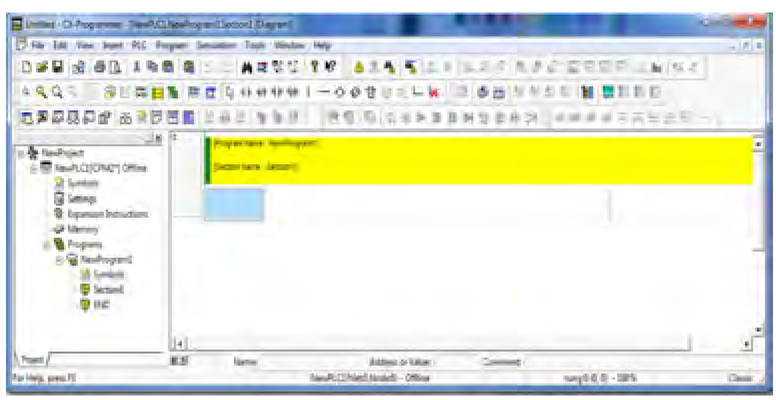
Gambar 52

6) Pada tampilan seperti gambar 4.5. Pilih tipe CPU yang digunakan dengan menekan tanda ▼ pada kotak ***Device Type*** (Perhatikan lingkaran pada gambar 4.5). Setelah itu akan muncul tampilan seperti gambar 4.6.

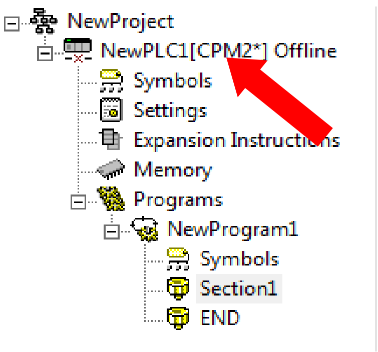


Gambar 53

7) Selanjutnya pilih tipe CPU yang digunakan, misal dalam kasus ini digunakan tipe CPM2. Untuk itu klik CPM2* (Perhatikan lingkaran merah pada gambar 4.6). Kemudian klik OK, akan muncul tampilan *ladder editor* atau bidang kerja (*worksheet*) seperti Gambar 4.7



Gambar 54

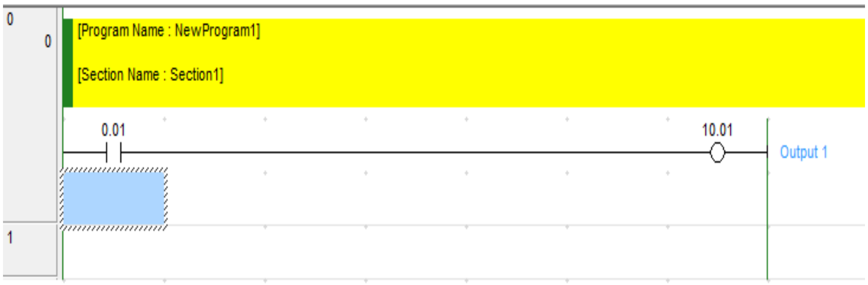


Gambar 55

Perhatikan "NewPLC1[CPM2*] Offline" pada bidang *project window* sebelah kiri layar (Perhatikan tanda panah pada Gambar 1.8) terlihat bahwa sudah tertera tipe CPU yang digunakan yaitu CPM2. Sekarang *ladder editor* atau bidang kerja (*worksheet*) sudah siap untuk membuat program.

b) Membuat Program

Sebagai contoh akan dibuat program seperti gambar 4.9 berikut ini.

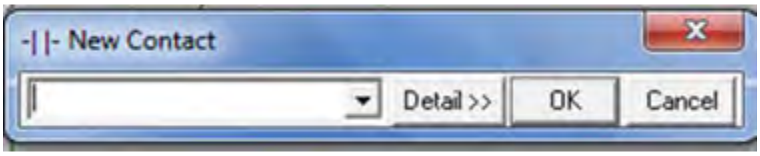


Gambar 56

Untuk membuat program seperti gambar 4.9 ikuti langkah-langkah sebagai berikut:

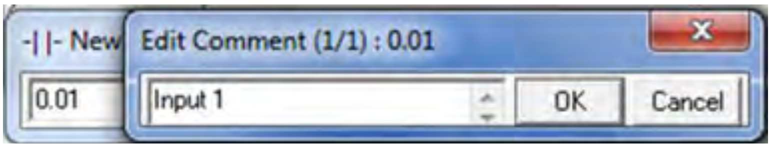
1) Langkah pertama adalah membuat kontak NO. Ada dua cara yang bisa ditempuh yaitu dengan mengklik simbol NO

(–I I–) pada *toolbar* atau dengan menekan huruf C pada *keyboard*. Setelah salah satu dari dua cara tersebut dilakukan akan muncul tampilan seperti gambar 4.10.



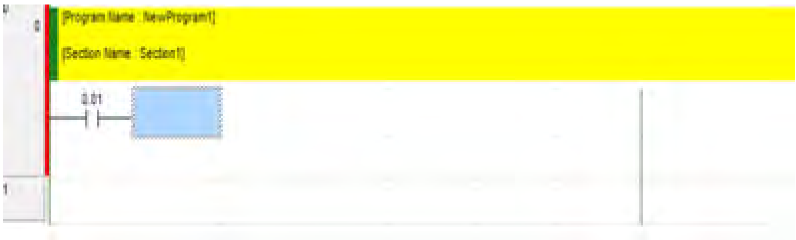
Gambar 57

2) Setelah muncul tampilan seperti gambar 4.10. Masukkan alamat input dengan format 0.01, setelah itu tekan enter atau klik OK dan ketikkan nama input pada kotak *Edit Comment* misal "Input 1" akan muncul tampilan seperti gambar 4.11.



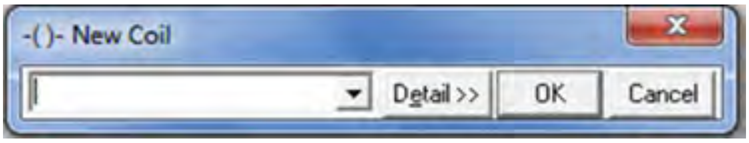
Gambar 58

3) Selanjutnya dari tampilan gambar 4.11 tekan enter atau klik OK, akan muncul tampilan seperti gambar 4.12. Terlihat sekarang bahwa ladder editor sudah terisi dengan input 0.01.



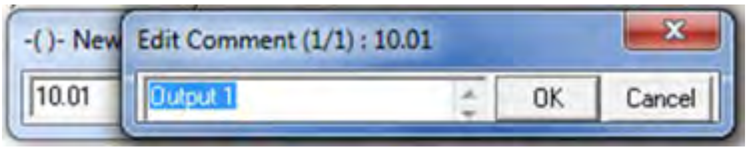
Gambar 59

4) Selanjutnya untuk memasukkan simbol output klik simbol output (-O-) pada *toolbar* atau tekan huruf O pada *keyboard*, muncul tampilan seperti gambar 4.13.



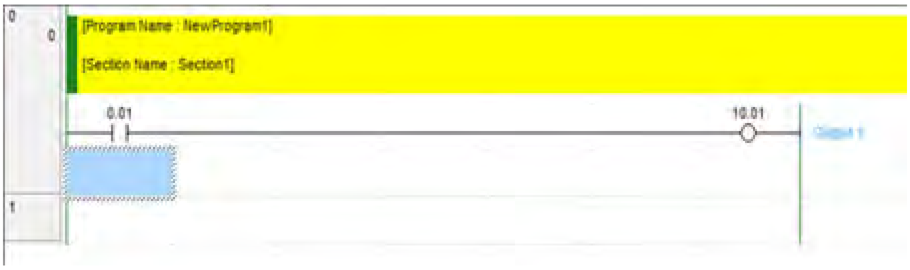
Gambar 60

5) Masukkan alamat output dengan format 10.01, kemudian tekan enter atau klik OK dan namai output, misal dengan "Output 1" (Perhatikan gambar 4.14).



Gambar 61

6) Kemudian tekan enter atau klik OK, akan muncul tampilan seperti gambar 4.15. Terlihat sekarang bahwa ladder editor sudah terisi dengan output 10.01.



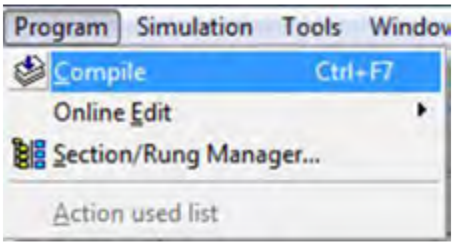
Gambar 62

Bila sudah muncul tampilan seperti gambar 4.15, berarti program sudah selesai dan siap dilanjutkan pada proses berikut.

c) Validasi Program

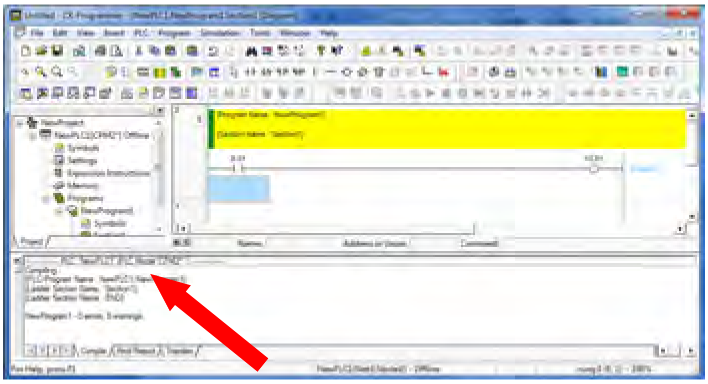
Langkah selanjutnya sebelum program bisa ditransfer dan dijalankan adalah melakukan validasi terhadap program yang sudah dibuat. Validasi program dilakukan dengan langkah-langkah sebagai berikut:

1) Pada *menubar* klik ***Program*** dan pilih ***Compile*** (perhatikan gambar 4.16)



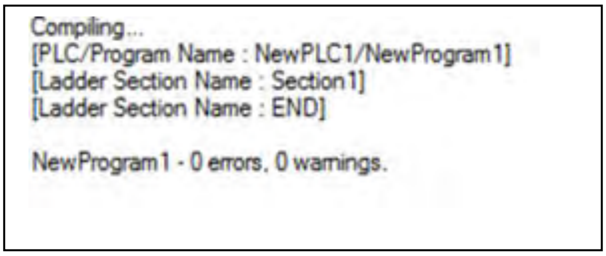
Gambar 63

2) Setelah diklik *Compile* akan muncul tampilan seperti gambar 4.17.



Gambar 64

3) Bila program tidak ada kesalahan, maka pada bagian bawah layar (perhatikan tanda panah pada gambar 4.17) muncul pesan seperti gambar 4.18


Compiling...
[PLC/Program Name : NewPLC1/NewProgram1]
[Ladder Section Name : Section1]
[Ladder Section Name : END]

NewProgram1 - 0 errors, 0 warnings.

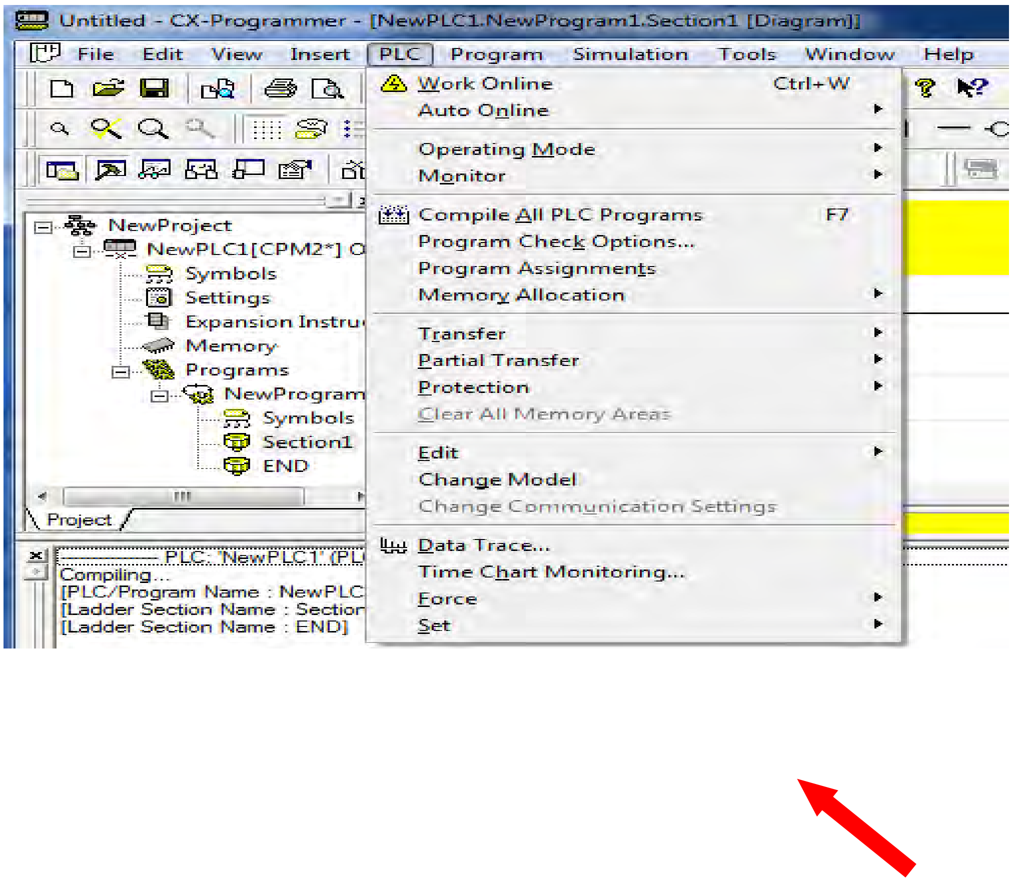
Gambar 65

4) Bila muncul pesan seperti ditunjukkan gambar 4.18 berarti program sudah "OK" dan siap untuk di transfer pada PLC.

d) Mentransfer Program

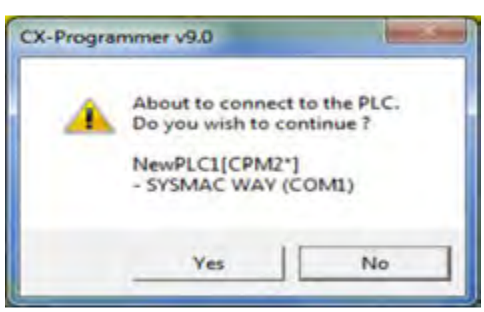
Untuk mentransfer program dari programmer dalam hal ini komputer ke PLC, ikuti langkah-langkah seperti berikut.

1) Klik menu ***PLC*** pada *toolbar* kemudian pilih ***Work Online*** (Perhatikan gambar 4.19)



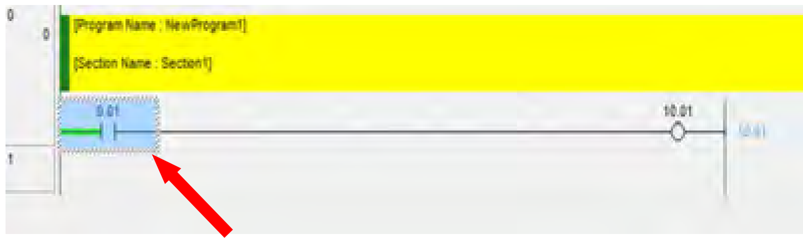
Gambar 66

2) Setelah diklik *Work Online* akan muncul pesan seperti gambar



Gambar 67

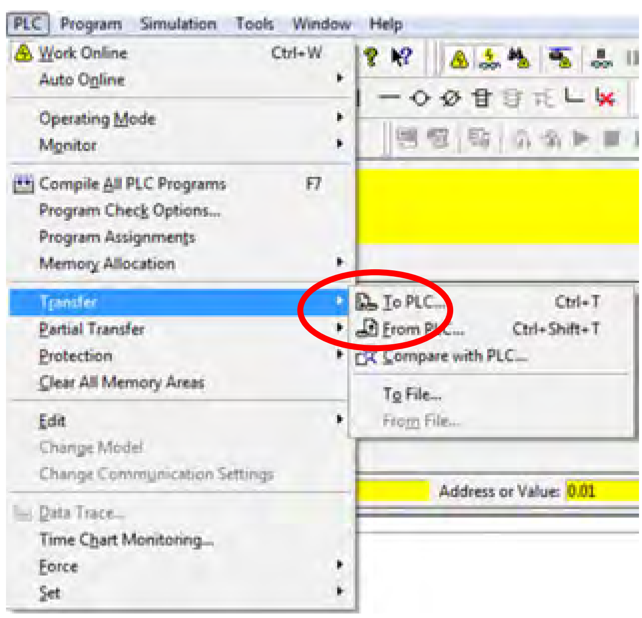
3) Konfirmasi dengan mengklik **Yes**, tampilan ladder editor akan berubah menjadi seperti gambar 4.21



Gambar 68

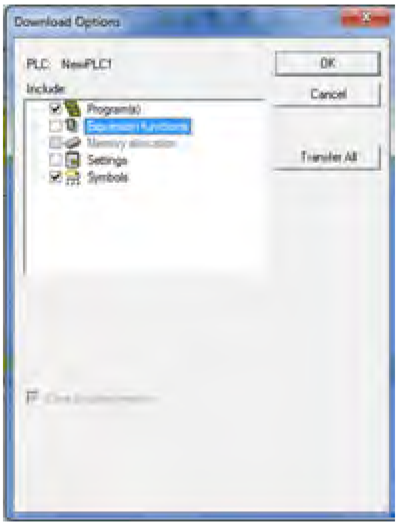
Perhatikan warna garis awal pada ladder diagram berubah jadi hijau. Ini berarti bahwa program sudah *online* dan siap di transfer ke PLC.

4) Untuk mentransfer program klik menu **PLC** dan pilih ***Transfer*** (Perhatikan gambar 4.22)



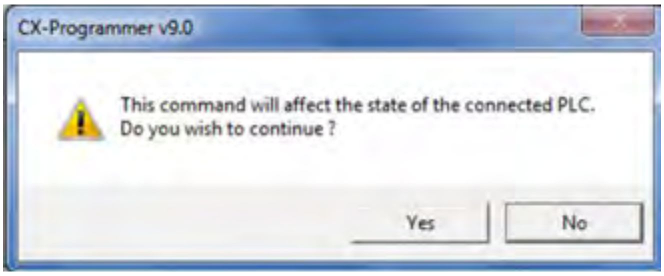
Gambar 69

5) Kemudian klik ***To PLC...*** (perhatikan lingkaran pada gambar 4.22, akan muncul tampilan seperti gambar 4.23



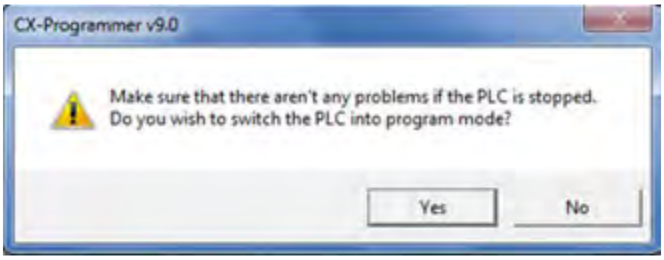
Gambar 70

6) Bila option sudah sesuai klik **OK**, akan muncul tampilan seperti gambar 4.24



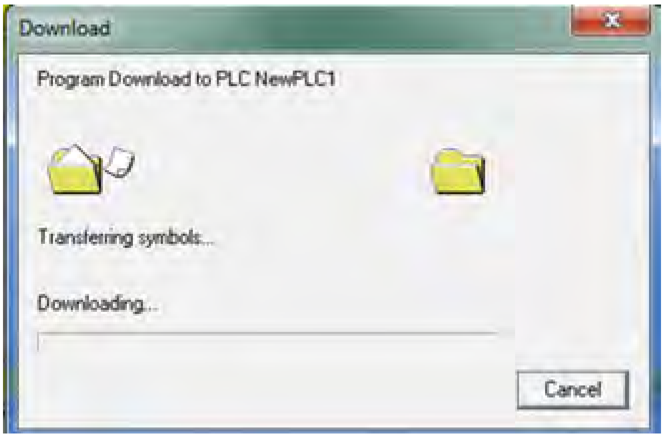
Gambar 71

7) Konfirmasi dengan mengklik Yes, akan muncul tampilan seperti gambar 4.25



Gambar 72

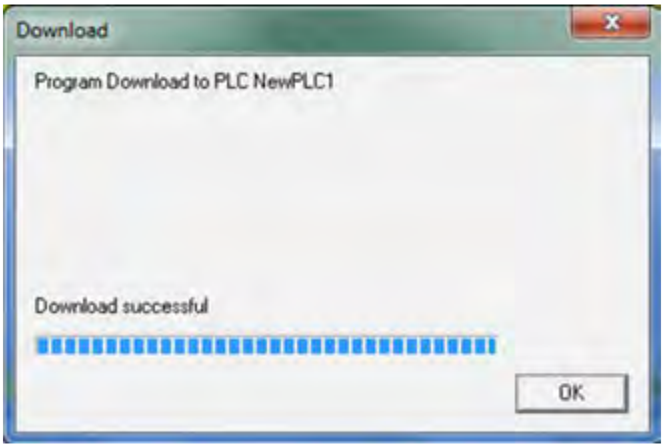
8) Kembali konfirmasi dengan mengklik **Yes**, akan muncul tampilan seperti gambar 4.26



Gambar 73

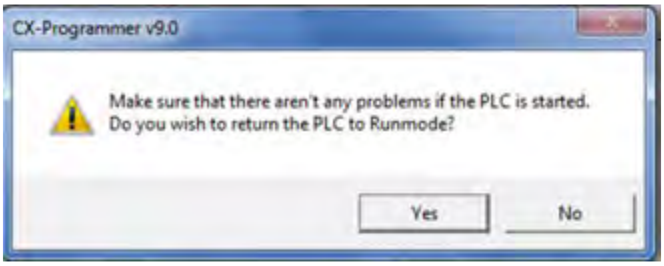
Gambar 4.26 di atas menunjukkan bahwa proses *transfer* atau *download* sedang berlangsung.

9) Bila proses transfer sukses, akan muncul tampilan seperti gambar 4.27



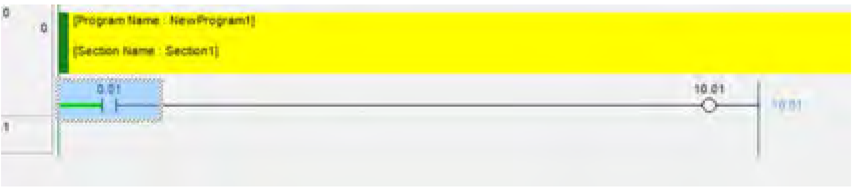
Gambar 74

10)Klik **OK**, akan muncul tampilan seperti gambar 4.28



Gambar 75

11)Konfirmasi dengan mengklik **Yes**, maka akan muncul tampilan ladder editor seperti ditunjukkan gambar 4.29



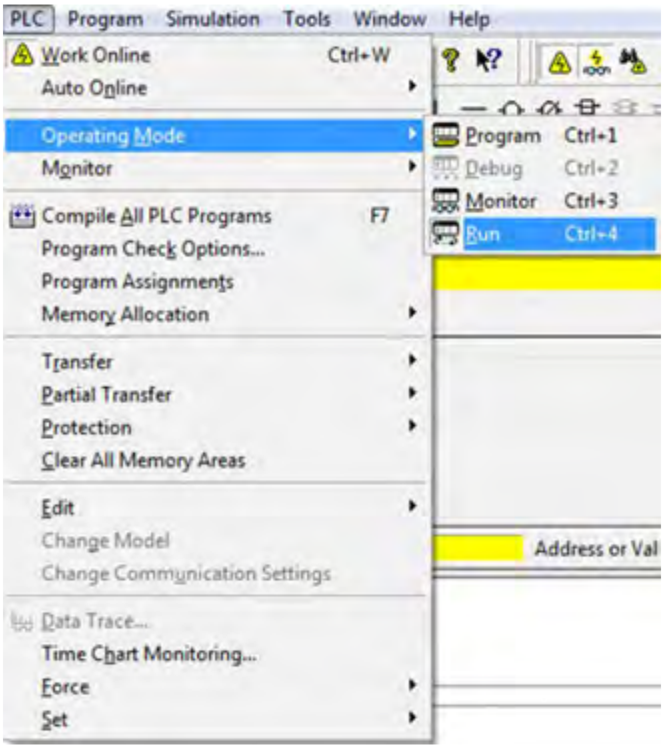
Gambar 76

12) Sekarang program sudah berada pada CPU PLC dan siap untuk dijalankan (*Run*)

e) Menjalankan Program

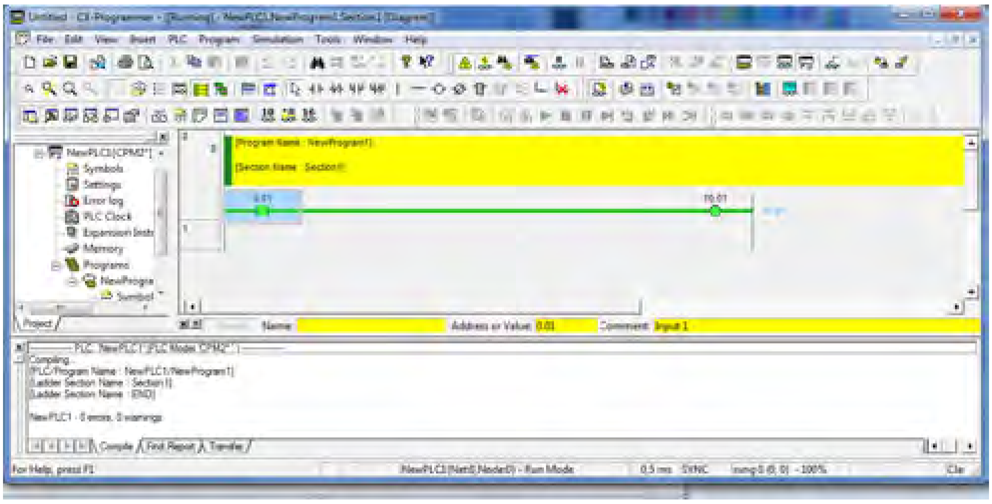
Untuk menjalankan program yang sudah berada pada PLC, ikuti langkah-langkah seperti berikut:

1) Klik **PLC** pada *menubar*, kemudian pilih **Operating Mode,** akan muncul tampilan seperti gambar 4.30. .



Gambar 77

2) Klik **Run** dan program siap dijalankan. Aktifkan input 0.01, maka output 10.01 akan aktif dan tampilan ladder editor akan berubah seperti ditunjukkan gambar 4.31.



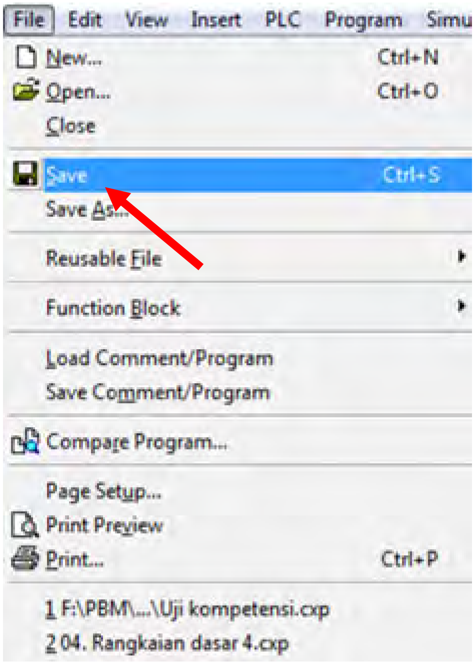
Gambar 78

Cermati gambar 4.31, sekarang output 10.01 dan baris program berubah warna menjadi hijau. Ini mengindikasikan bahwa baris program tersebut *conductive* atau ON (bit 1).

f) Menyimpan Program

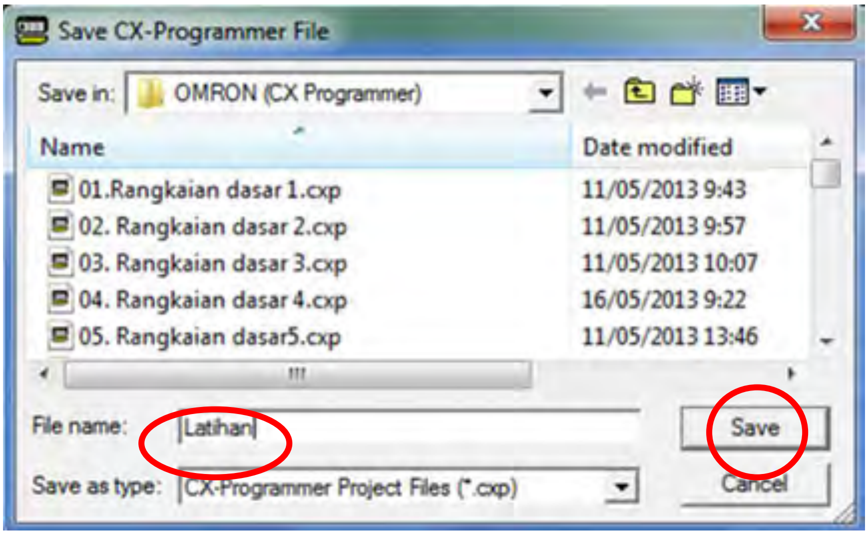
Untuk menyimpan program dapat dilakukan dengan cara sebagai berikut:

1. Klik "***File***" pada *menubar*, kumudian pilih "***Save***" (perhatikan tanda panah pada gambar 4.32).



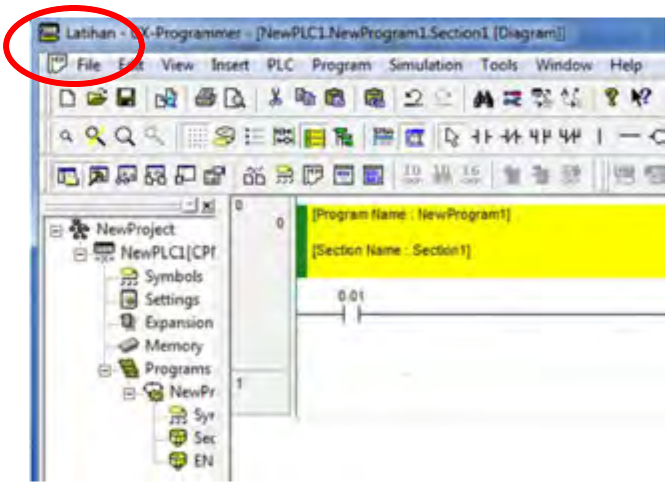
Gambar 79

2. Klik "*Save*" akan muncul tampilan seperti gambar 4.33.



Gambar 80

3. Pilih "***drive***" untuk menyimpan program kemudian tempatkan *cursor* pada kotak "*File name*" dan ketikkan nama file, misal Latihan (perhatikan lingkaran pada gambar 4.33). Kemudian klik "Save", tampilan ladder editor akan berubah seperti ditunjukkan gambar 4.34.



Gambar 81

Perhatikan lingkaran pada gambar 4.34 di atas, sekarang terlihat bahwa sudah tercantum nama file yaitu "Latihan". Ini sebagai indikator bahwa program sudah tersimpan pada komputer.

g) Mengedit Program

Misalkan kita mempunyai program seperti ditunjukkan gambar 4.35. Pada program tersebut terdapat beberapa kesalahan antara lain:

Input 0.00 mestinya adalah 0.01

- Output 10.11 mestinya adalah 10.01
- Input 0.03 mestinya adalah kontak dari output 10.11
- Setelah input 0.02 mestinya terdapat kontak NC input 0.03

untuk sensor proteksi.

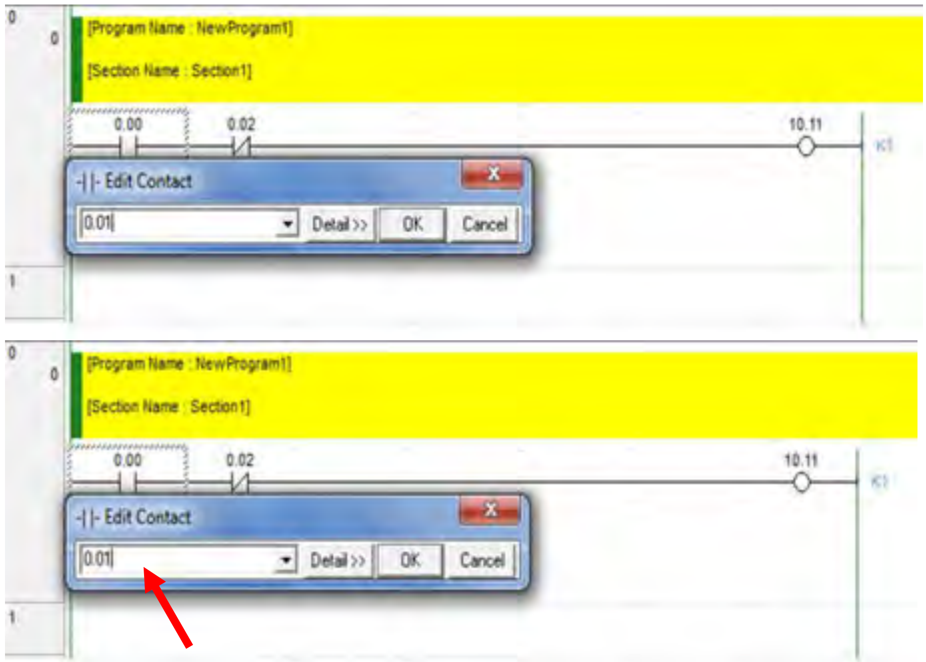


Gambar 82

Untuk mengedit atau memperbaiki kesalahan tersebut, lakukan dengan langkah-langkah sebagai berikut.

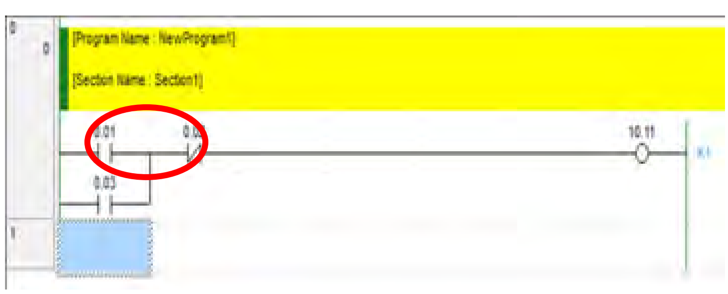
Mengganti *address* input 0.00 menjadi 0.01

1. Tempatkan *cursor* pada kontak 0.00, kemudian *double click* akan muncul tampilan seperti gambar 4.35.



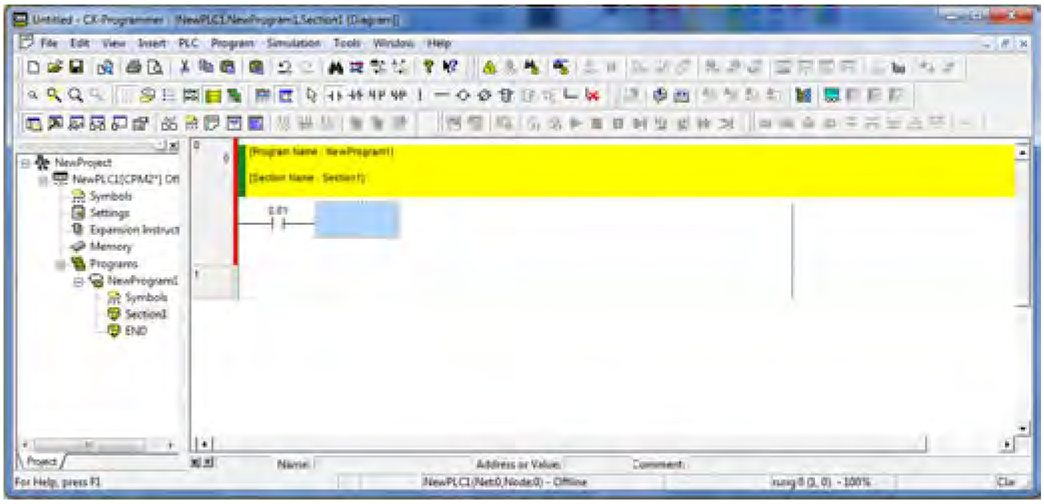
Gambar 83

2. Tempatkan *cursor* pada kotak "***Edit Contact***" kemudian ganti 0.00 dengan 0.01 (perhatikan lingkaran pada gambar 4.36) dan kilik Ok atau tekan enter, muncul kotak "Edit Comment", tekan kembali enter atau klik OK, maka akan muncul tampilan seperti gambar 4.37. Perhatikan input 0.00 sekarang sudah berubah jadi 0.01.



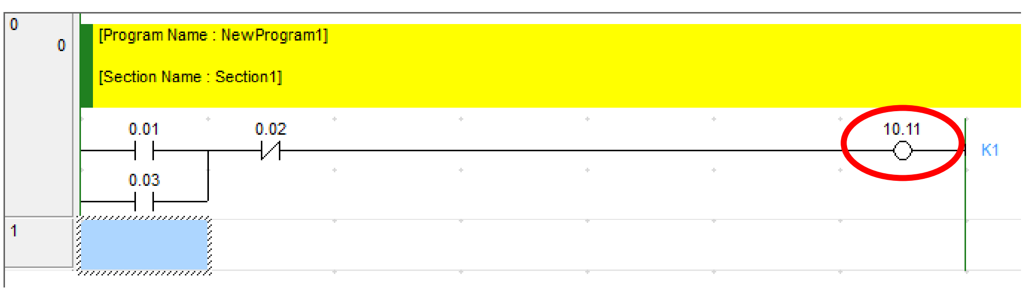
Gambar 84

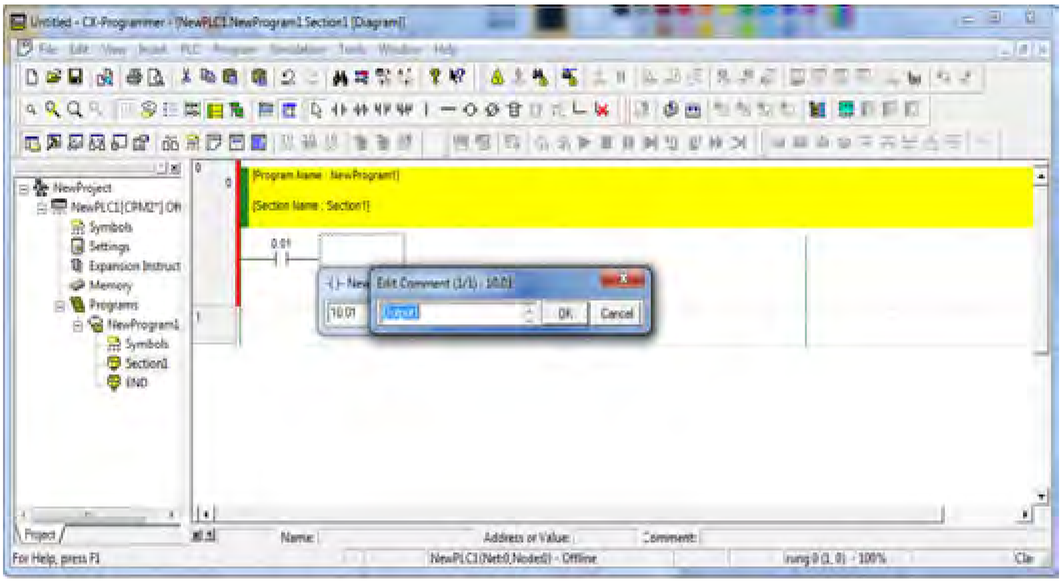
Gambar 85

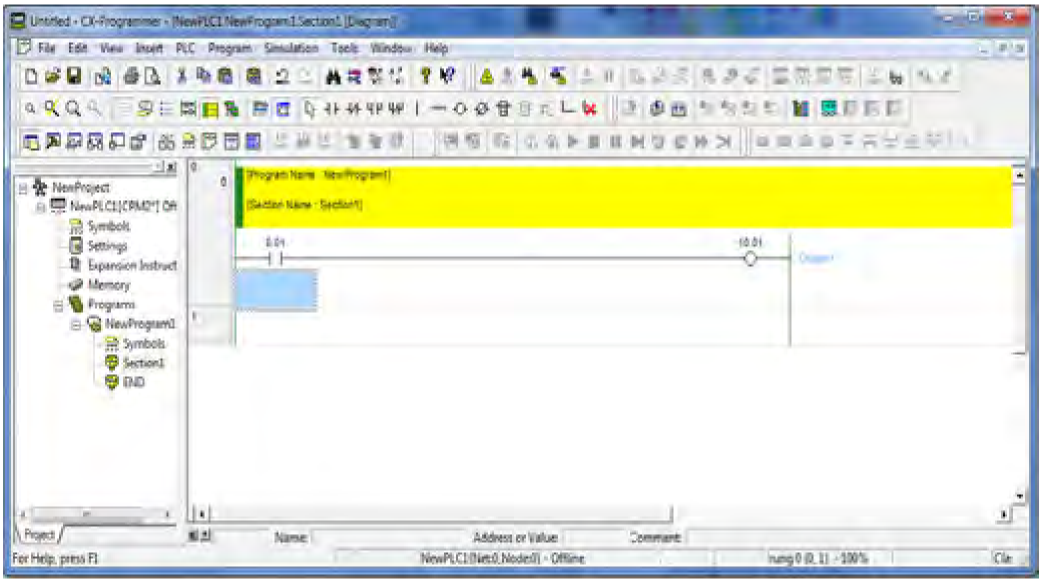


Gambar 86

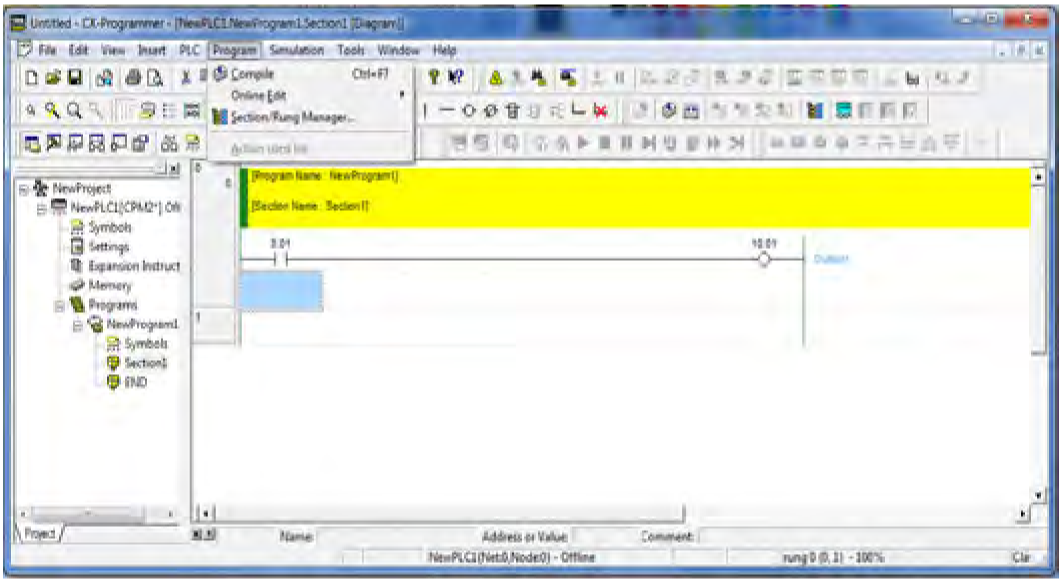
1. Mengganti address output 10.11 menjadi 10.01.Tempatkan *cursor* pada kotak "***Edit Contact***" kemudian ganti 10.00 dengan 10.01 (perhatikan lingkaran pada gambar 4.38) dan kilik Ok atau tekan enter, muncul kotak "Edit Comment", tekan kembali enter atau klik OK, maka akan muncul tampilan seperti gambar 4.39. Perhatikan input 10.11 sekarang sudah berubah jadi 10.01.

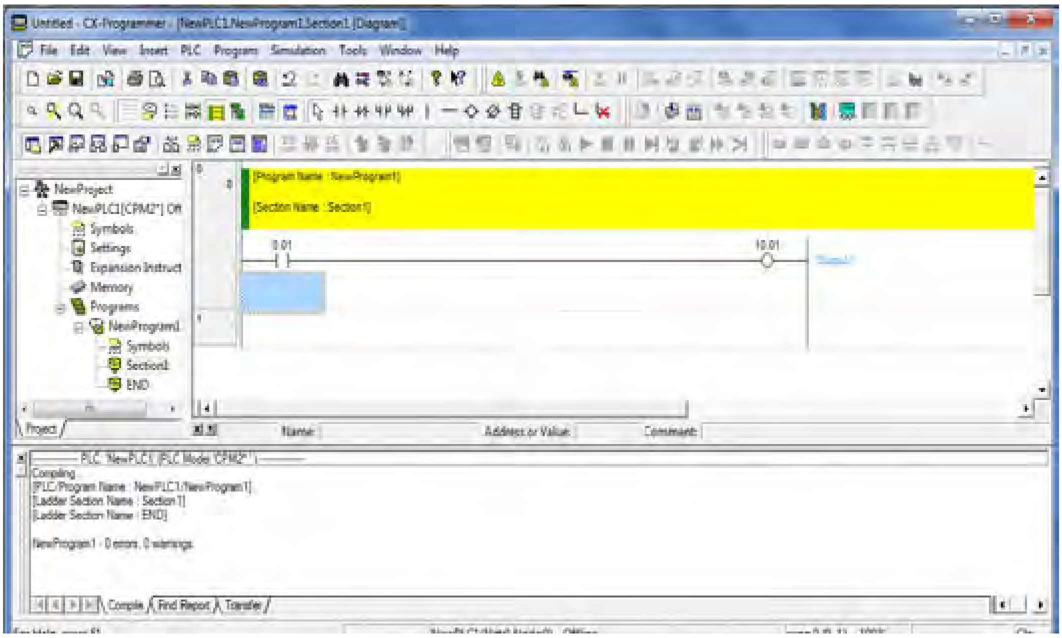






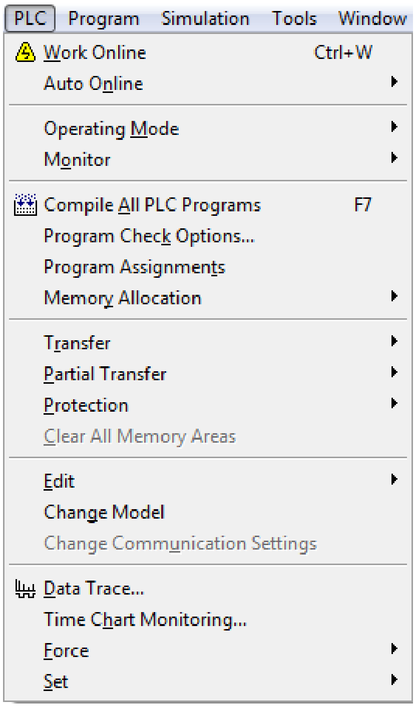
Gambar 87

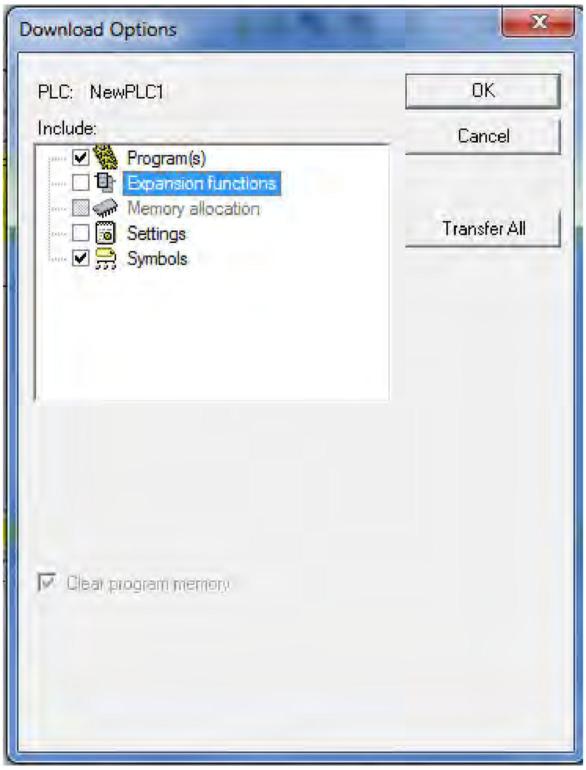


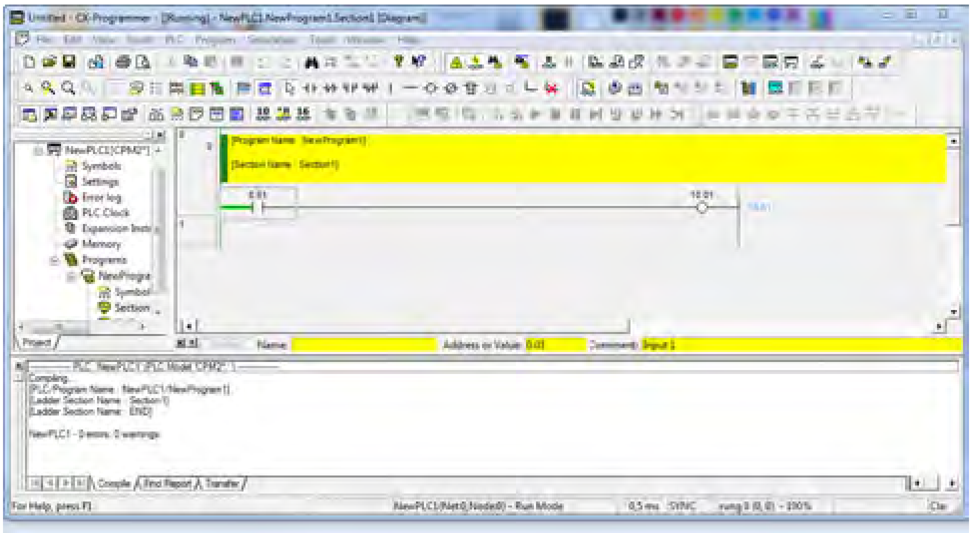


---------------- PLC: 'NewPLC1' (PLC Model 'CPM2* ') ----------------
Compiling...
[PLC/Program Name : NewPLC1/NewProgram1]
[Ladder Section Name : Section1]
[Ladder Section Name : END]

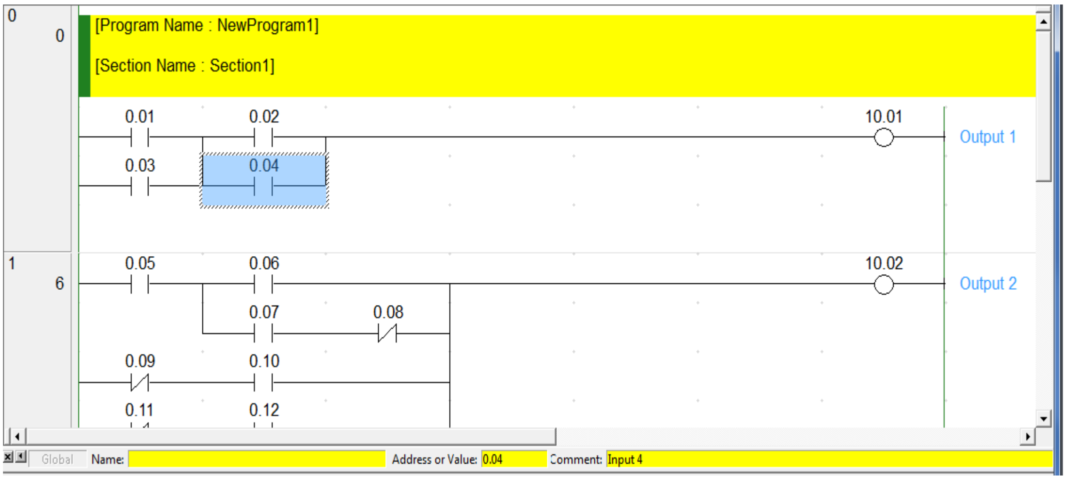
NewProgram1 - 0 errors, 0 warnings.

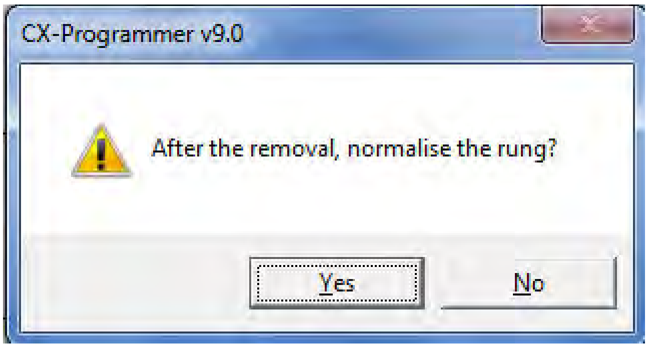

PLC
Program
Simulation
Tools
Window
Work Online Ctrl+W
Auto Online
Operating Mode
Monitor
Compile All PLC Programs F7
Program Check Options...
Program Assignments
Memory Allocation
Transfer
Partial Transfer
Protection
Clear All Memory Areas
Edit
Change Model
Change Communication Settings
Data Trace...
Time Chart Monitoring...
Force
Set

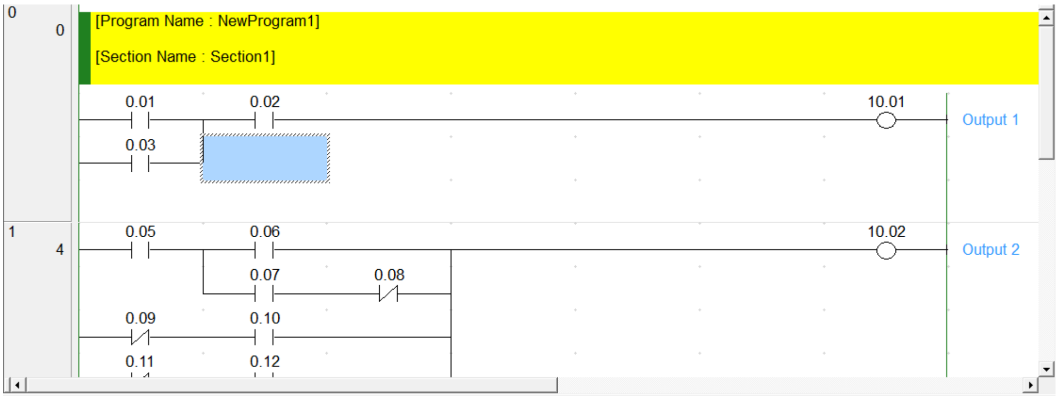

Download Options
PLC: NewPLC1
OK
Include:
Cancel
Program(s)
Expansion functions
Memory allocation
Settings
Symbols
Transfer All
Clear program memory

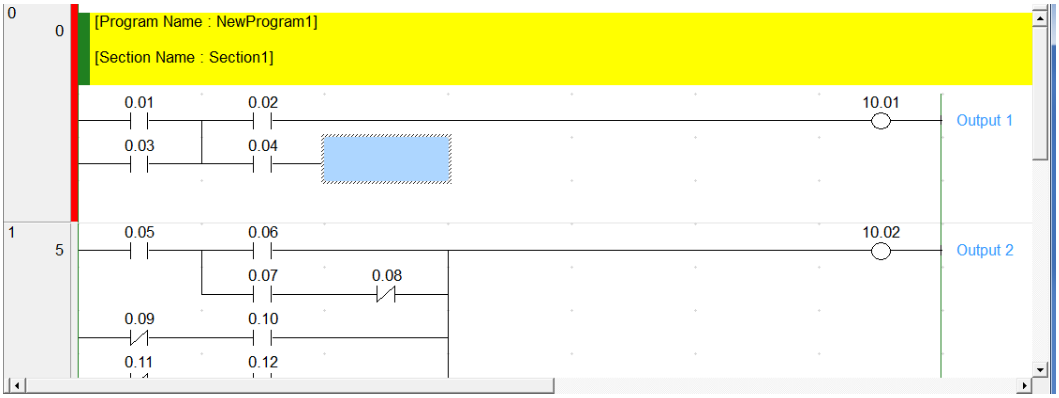


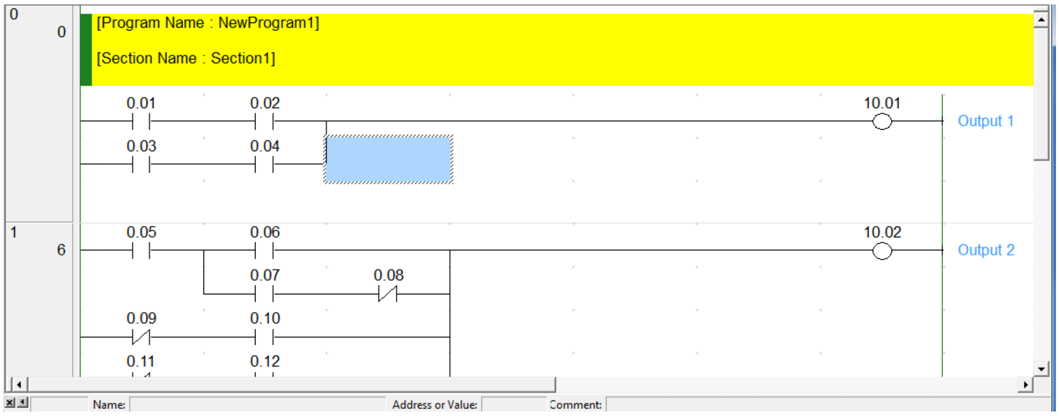
Mengedit Program



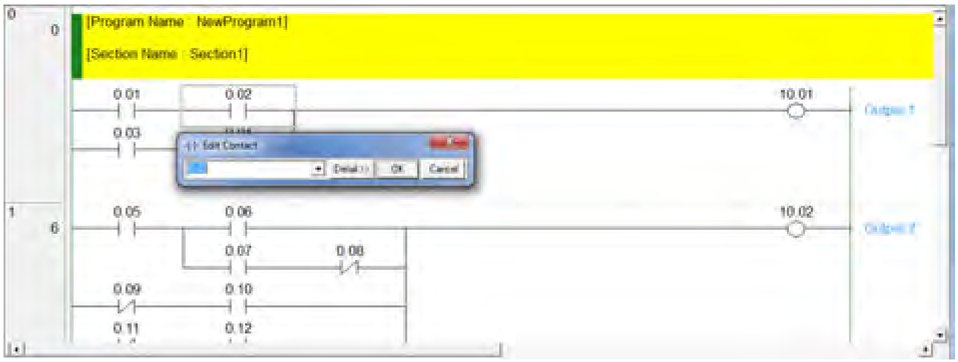


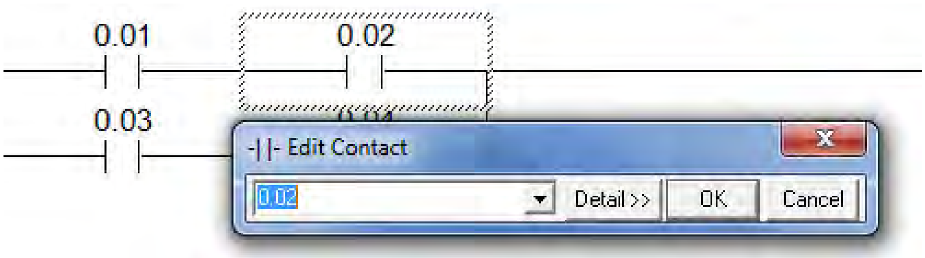

[Program Name : NewProgram1]
[Section Name : Section1]
0.01
0.02
10.01
Output 1
0.03
0.05
0.06
10.02
Output 2
0.07
0.08
0.09
0.10
0.11
0.12

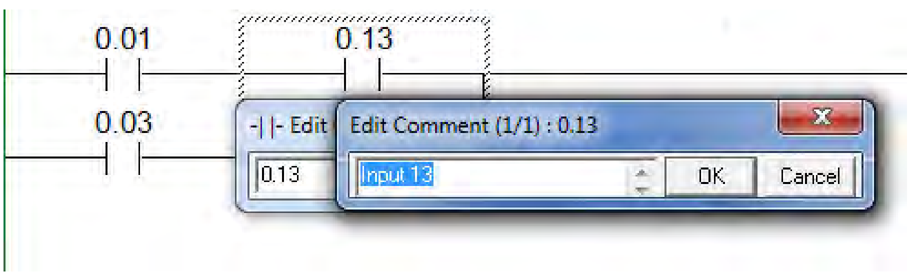

[Program Name : NewProgram1]
[Section Name : Section1]
0.01
0.02
10.01
Output 1
0.03
0.04
0.05
0.06
10.02
Output 2
0.07
0.08
0.09
0.10
0.11
0.12


[Program Name : NewProgram1]
[Section Name : Section1]
0.01
0.02
10.01
Output 1
0.03
0.04
0.05
0.06
10.02
Output 2
0.07
0.08
0.09
0.10
0.11
0.12
Name:
Address or Value:
Comment:

Merubah address kontak

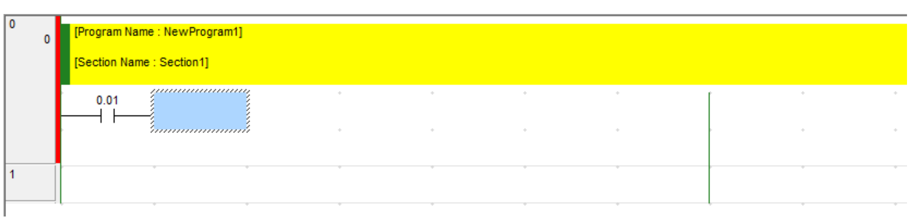






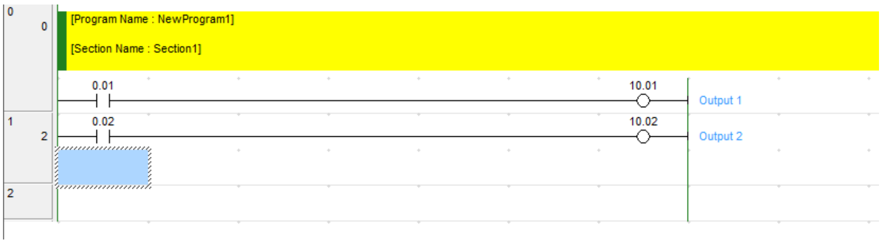
Memprogram berbagai jenis output

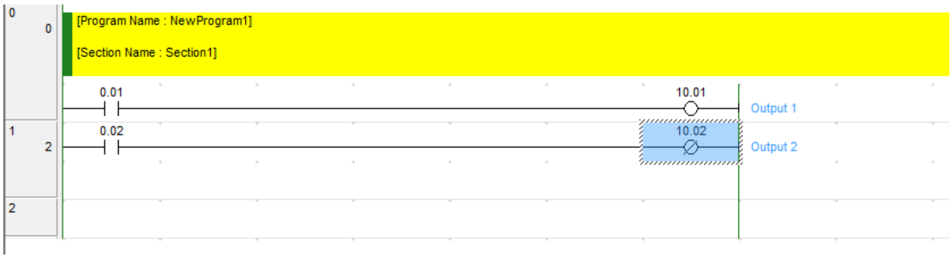
1. Output Direct



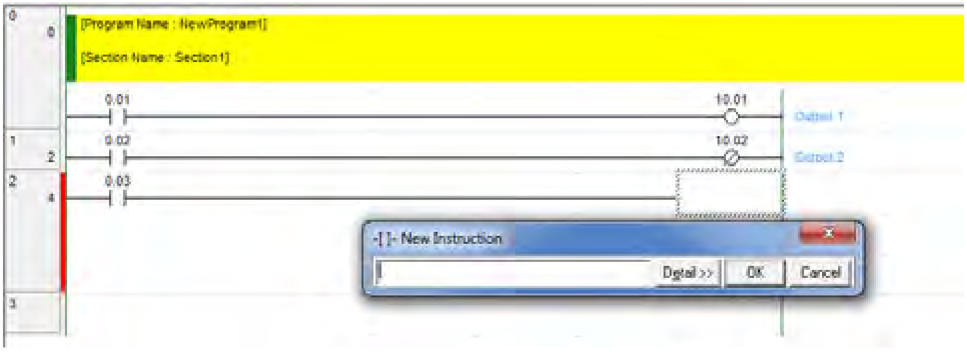


2. Output Inverse

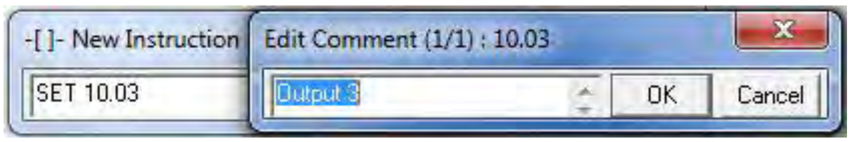


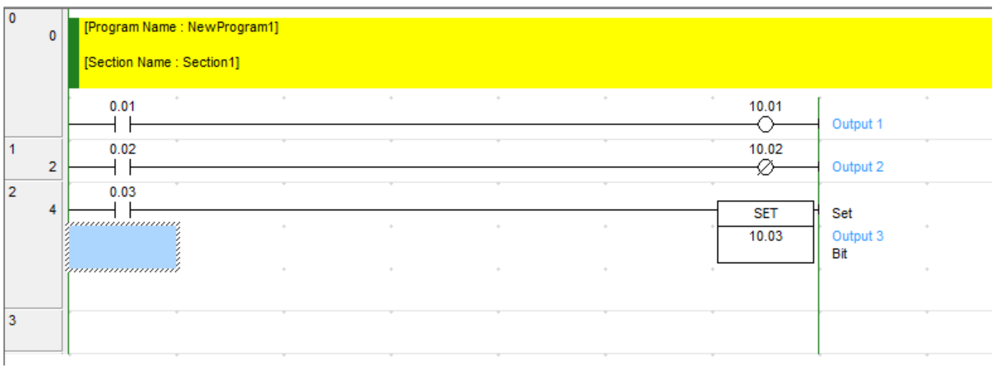


3. Output latched

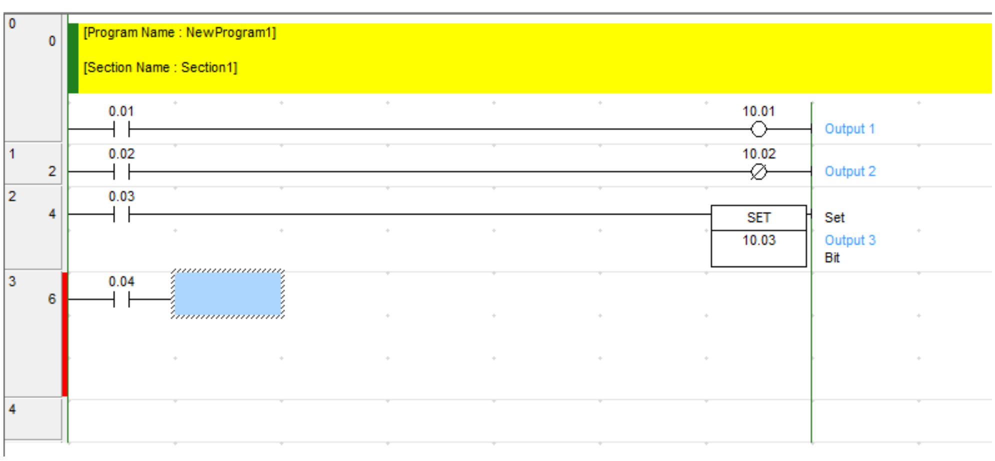


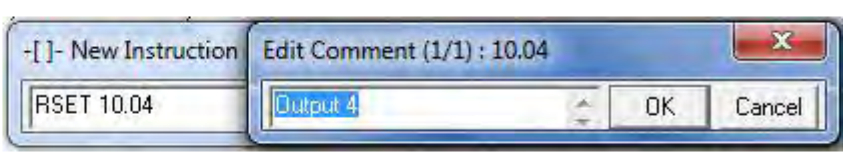


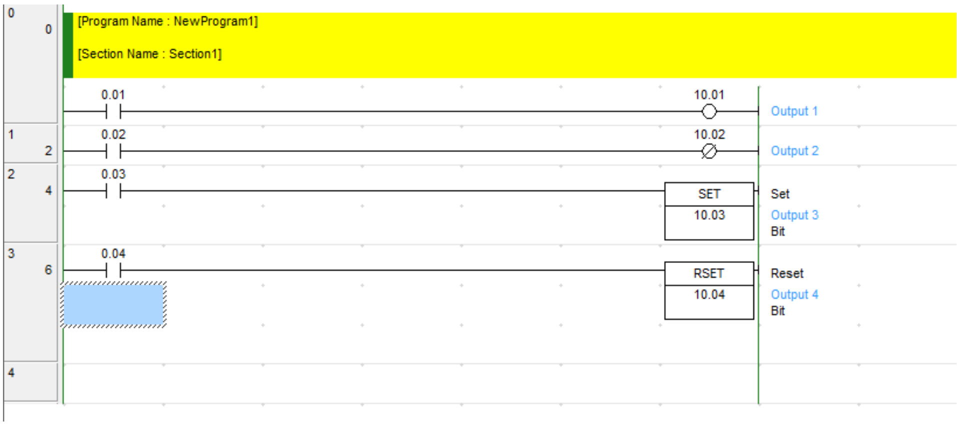




4. Output Unlatched

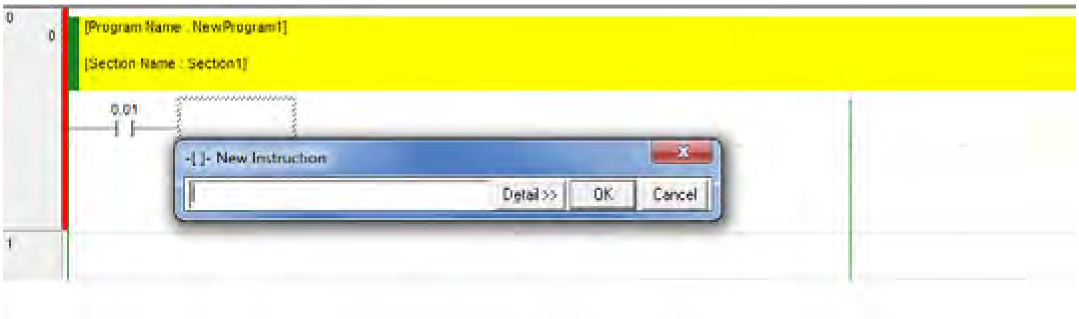


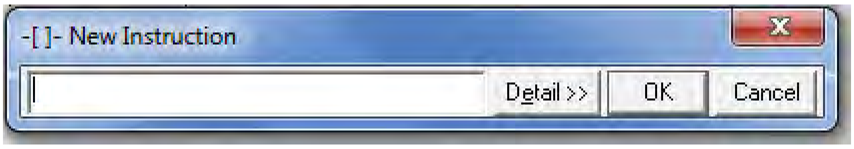




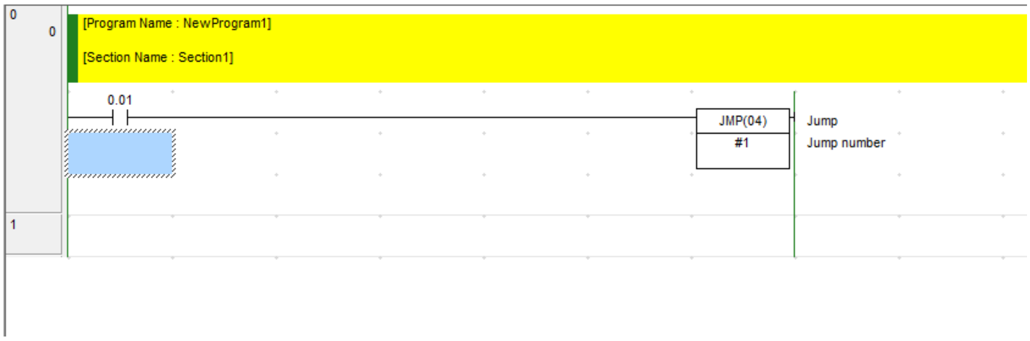
5. Output jump

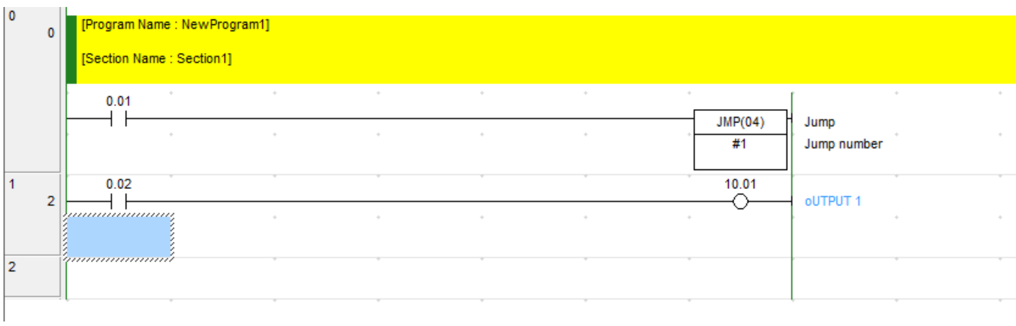




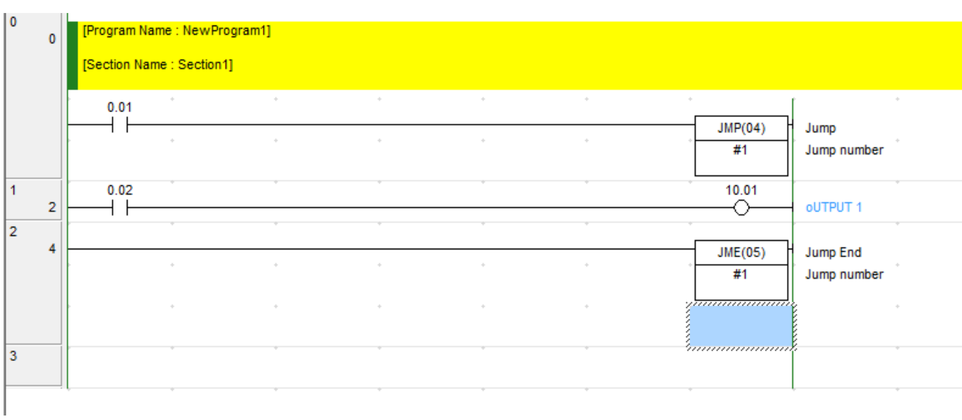




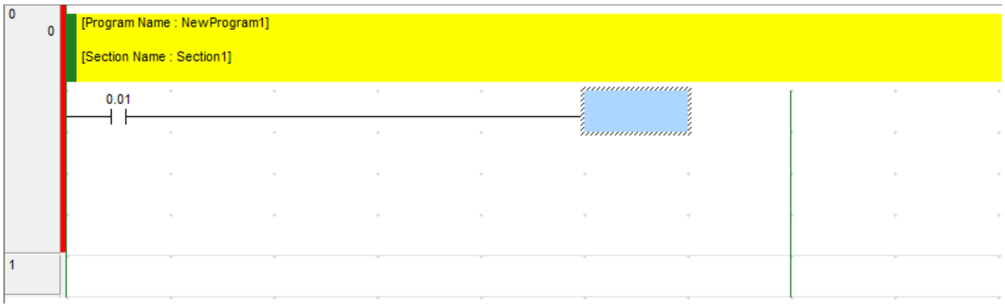

[Program Name : NewProgram1]
[Section Name : Section1]
0.01
JMP(04)
#1
Jump
Jump number


[Program Name : NewProgram1]
[Section Name : Section1]
0.01
JMP(04)
#1
Jump
Jump number
0.02
10.01
oUTPUT 1


-[ ]- New Instruction
JME #1
Detail >>
OK
Cancel

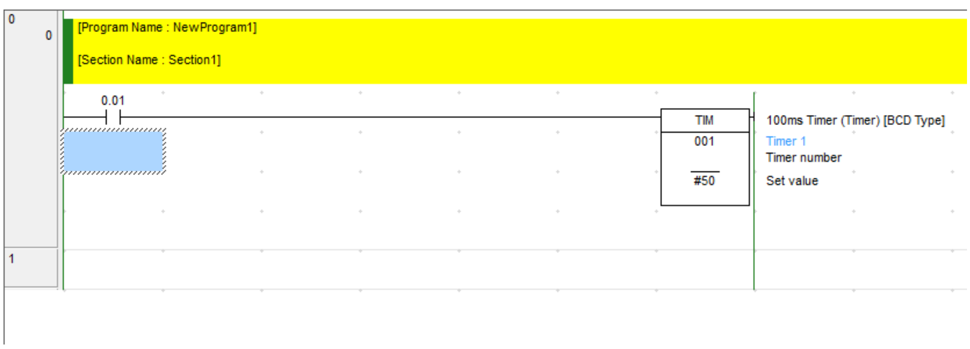

[Program Name : NewProgram1]
[Section Name : Section1]
0.01
JMP(04)
#1
Jump
Jump number
0.02
10.01
oUTPUT 1
JME(05)
#1
Jump End
Jump number

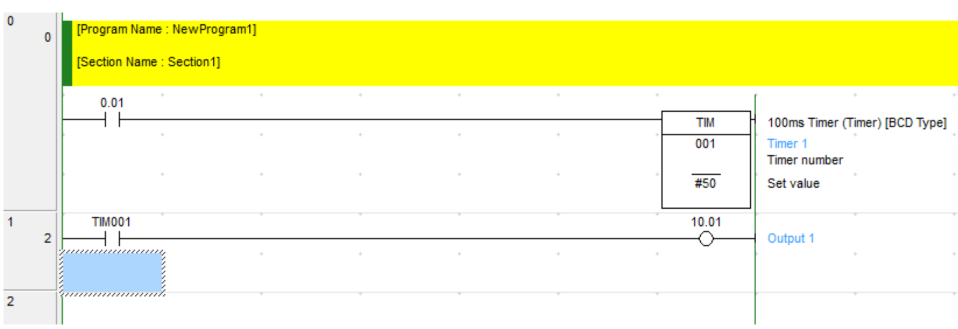
Memprogram Timer



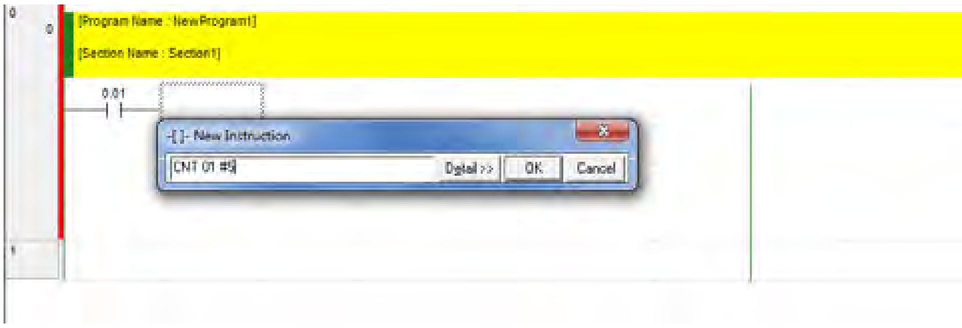




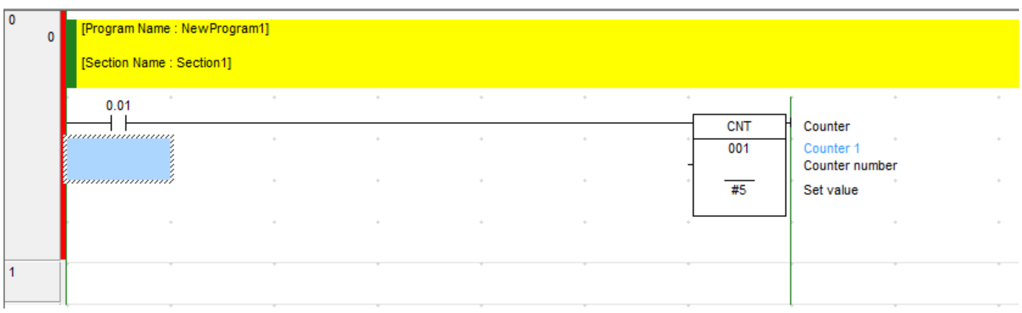


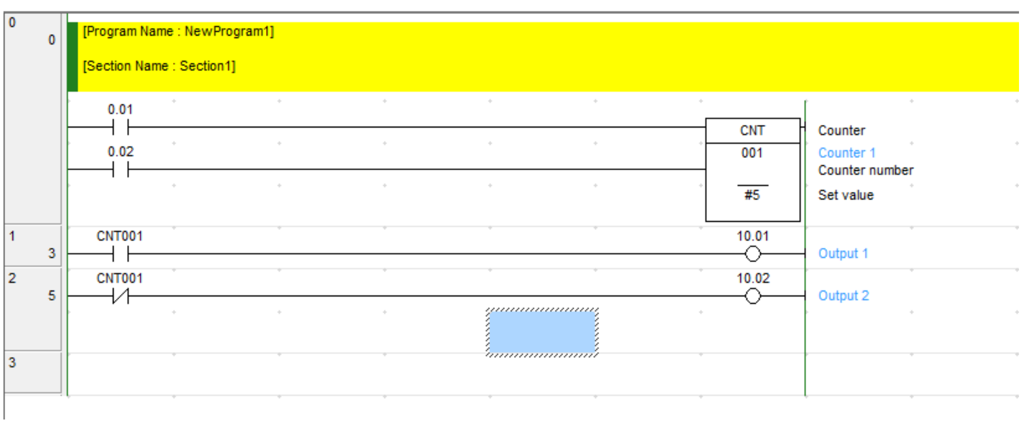


Memprogram Counter



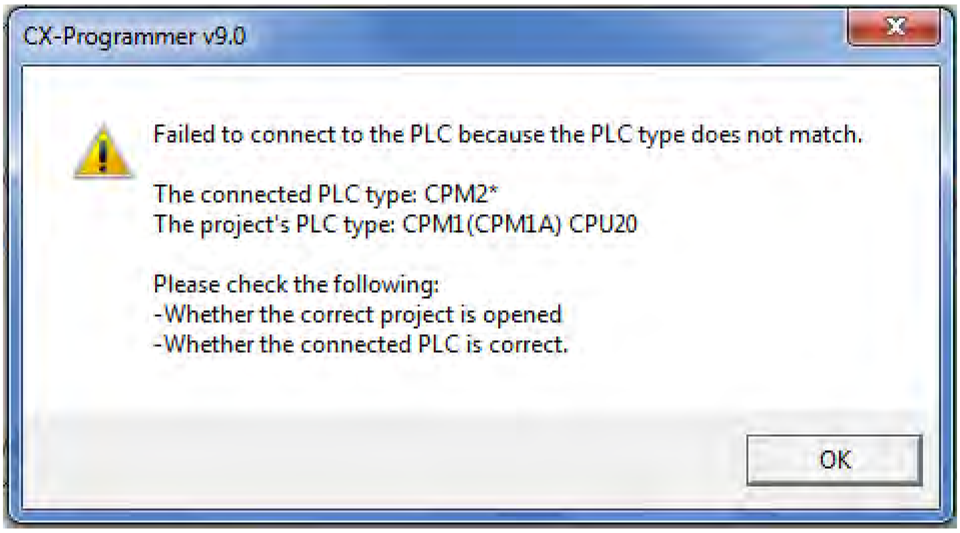




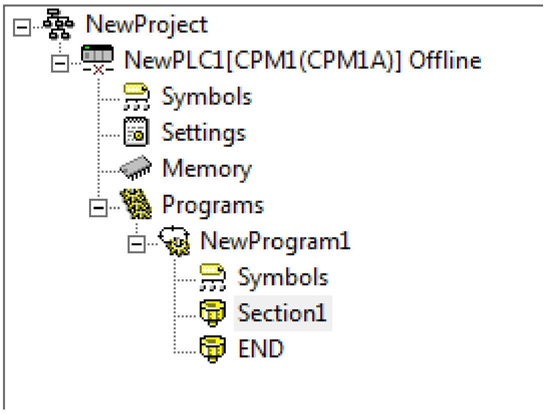


Merubah Setting CPU dan Komunikasi

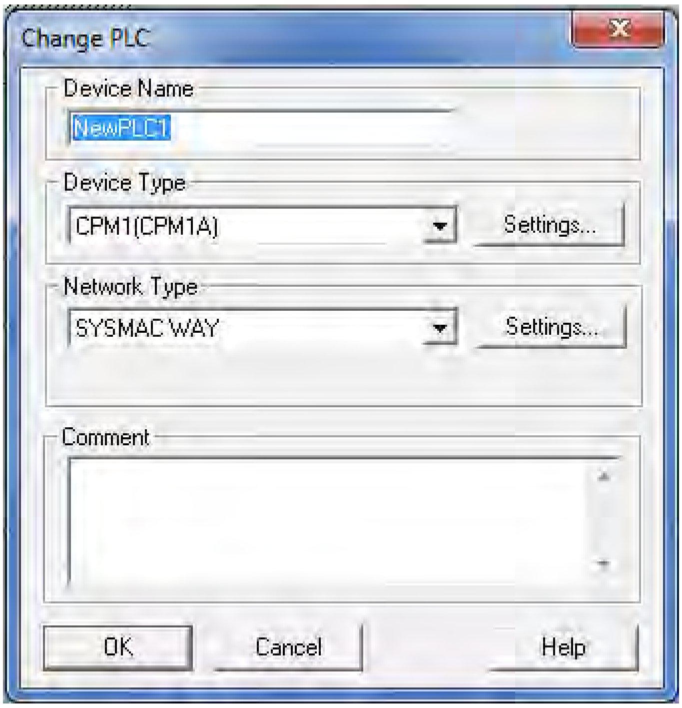
Bila seting tipe CPU program tidak bersesuaian dengan tipe CPU yang digunakan, maka akan muncul pesan berikut ini saat di klik "work online"



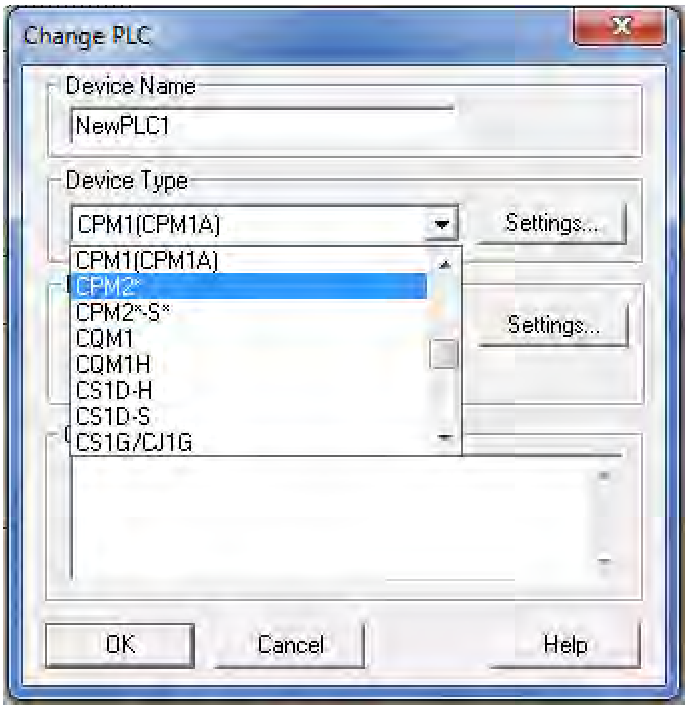
Untuk itu perlu dilakukan perubahan seting tipe CPU dengan cara sebagai berikut:



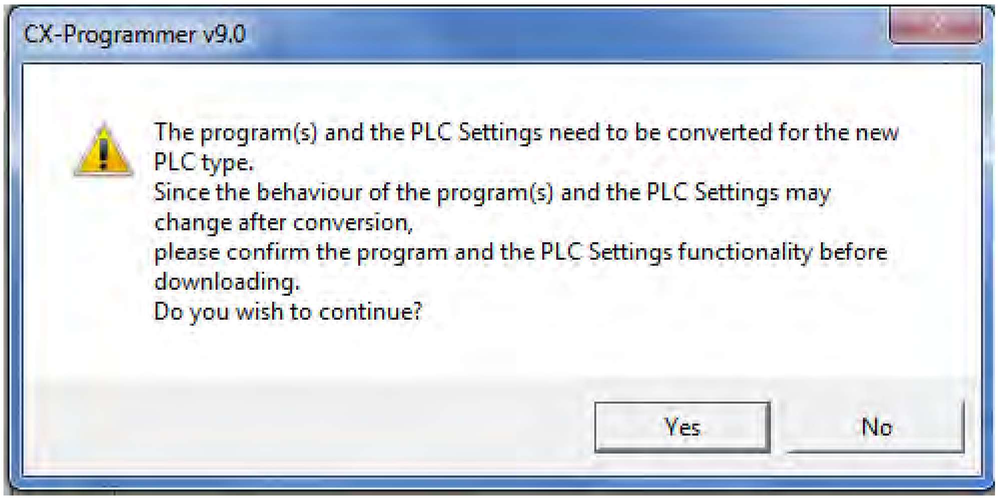
Klik "NewPLC1[CPM1(CPM1A)] Offline" pada *project manager* yang terdapat pada sebelah kanan atas layar dua kali, maka akan muncul tampilan sebagai berikut:



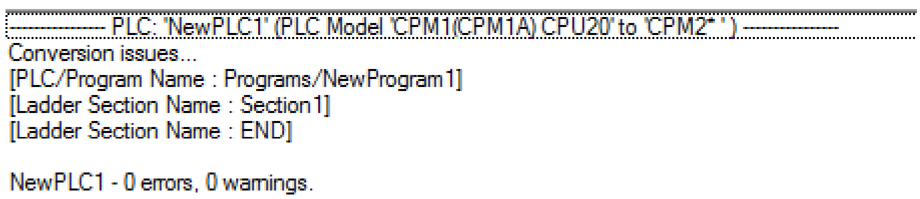
Klik tanda ▼ pada Device Type, kemudian pilh tipe CPU (dalam contoh ini CPM2*) kemudian klik OK



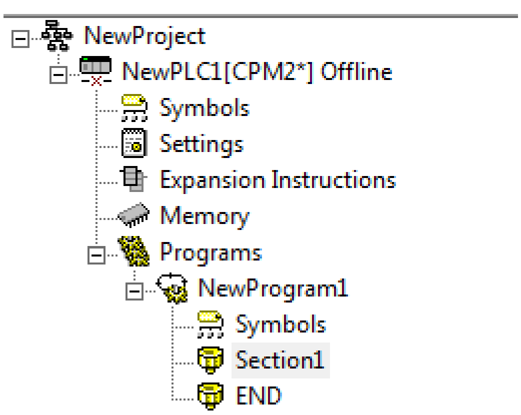
Setelah diklik OK akan muncul tampilan sebagai berikut dan klik YES



Bila konversi berhasil maka akan muncul tampilan seprti berikut pada bagian bawah layar



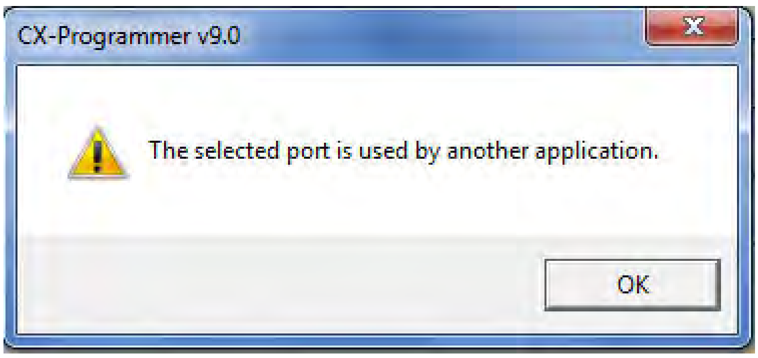
Tampilan project manager berubah seperti berikut:



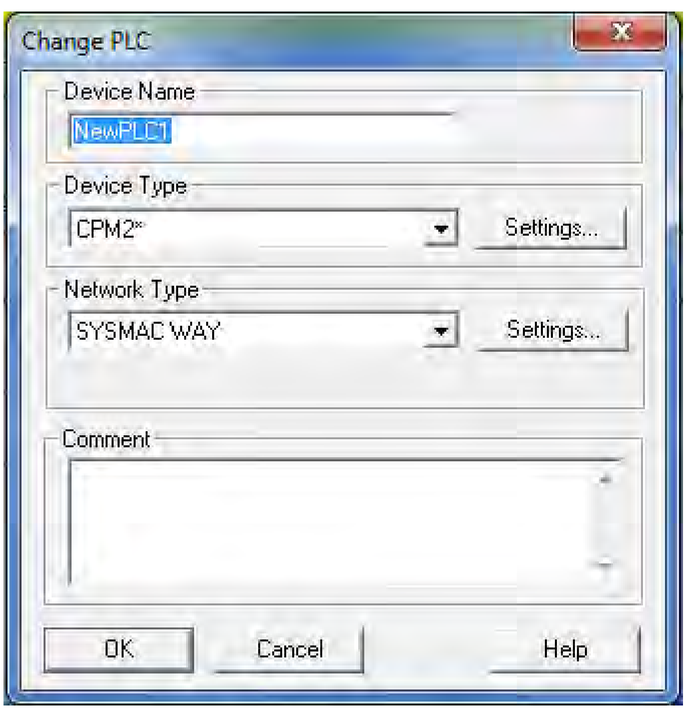
Perhatikan tipe CPU pada project manager di atas, sekarang sudah berubah menjadi CPM2*

Bila sudah muncul pesan seperti di atas, artinya program sudah siap untuk dihubungkan dengan CPU PLC dan ditransfer untuk kemudian dijalankan dengan seting yang baru.

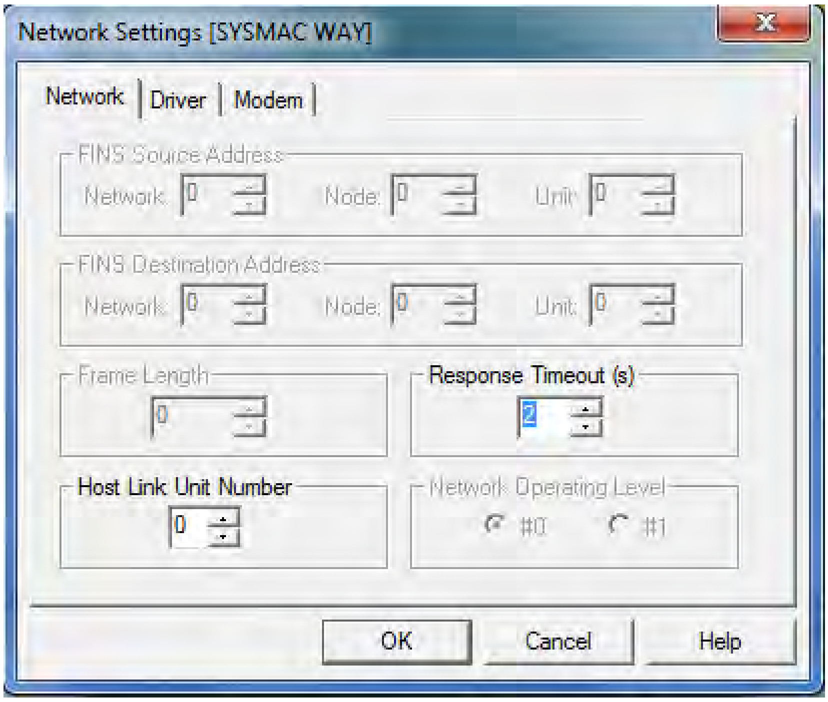
Perubahan seting juga perlu dilakukan bilamana port USB komputer tidak bersesuaian dengan seting pada program. Pesan yang muncul pada saat program dikoneksi adalah seperti berikut:



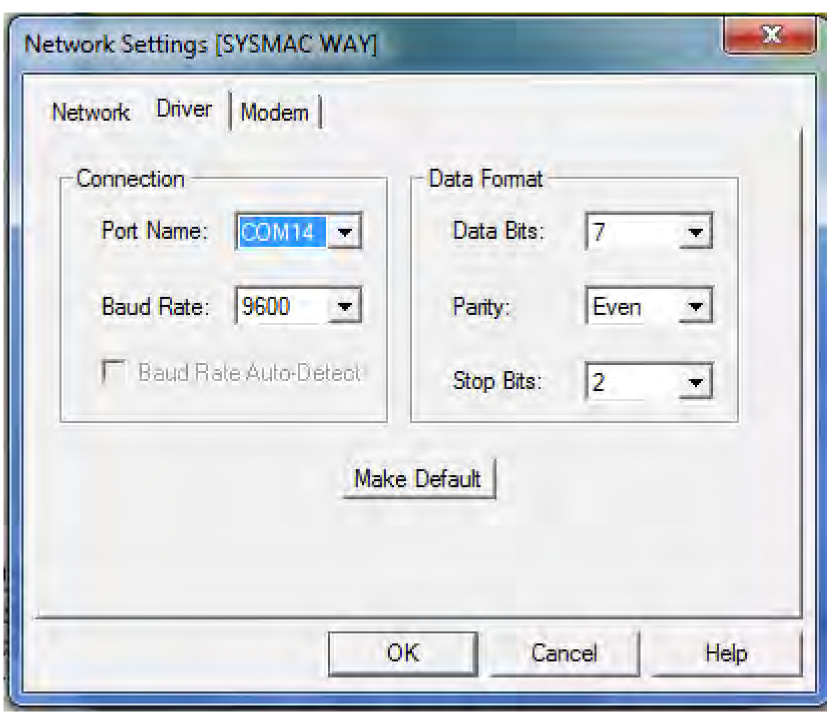
Klik OK dan double klik "NewPLC1[CPM1(CPM1A)] Offline" pada *project manager,* akan muncul tampilan seperti berikut:



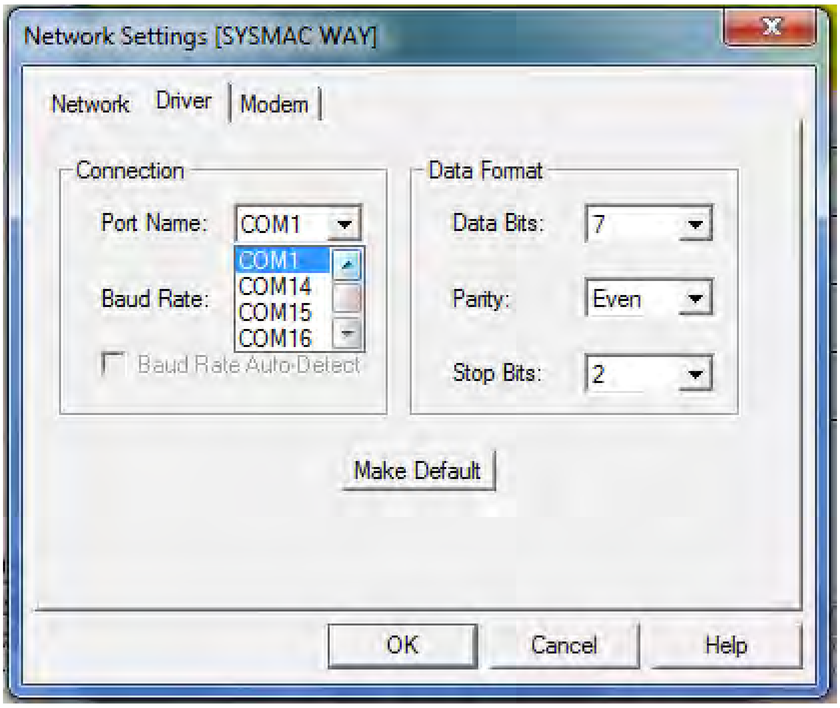
Klik “Setting...”, akan muncul tampilan seperti berikut:



Dari tmapilan di atas klik “Driver”, akan muncul tampilan seperti berikut:

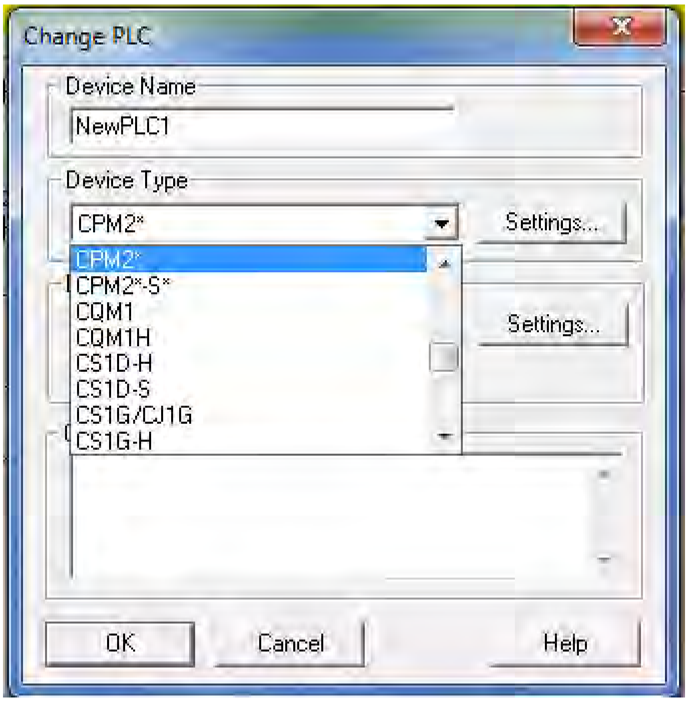


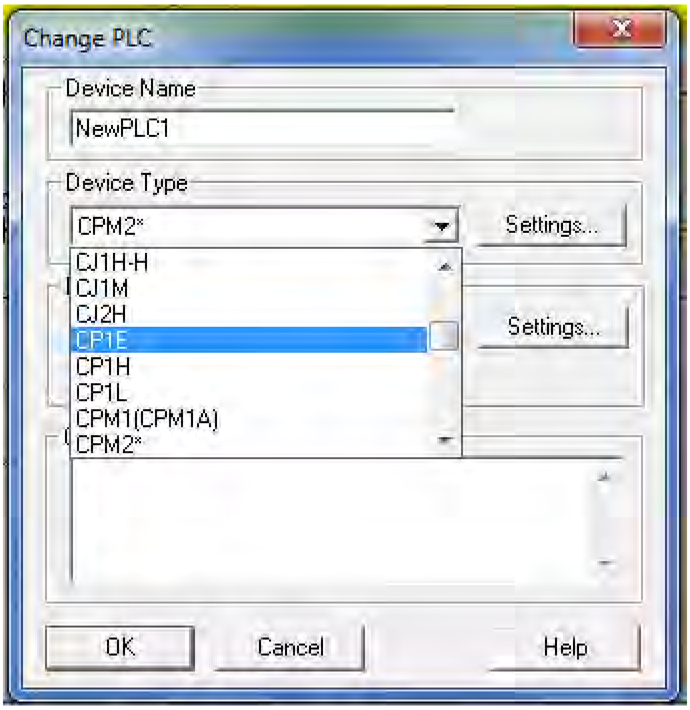
Dari tampilan di atas klik ▼ Port Name dan pilih port yang sesuai dengan seting komputer. Contoh seting port komputer nya COM1 maka pilih COM1 dan klik OK.

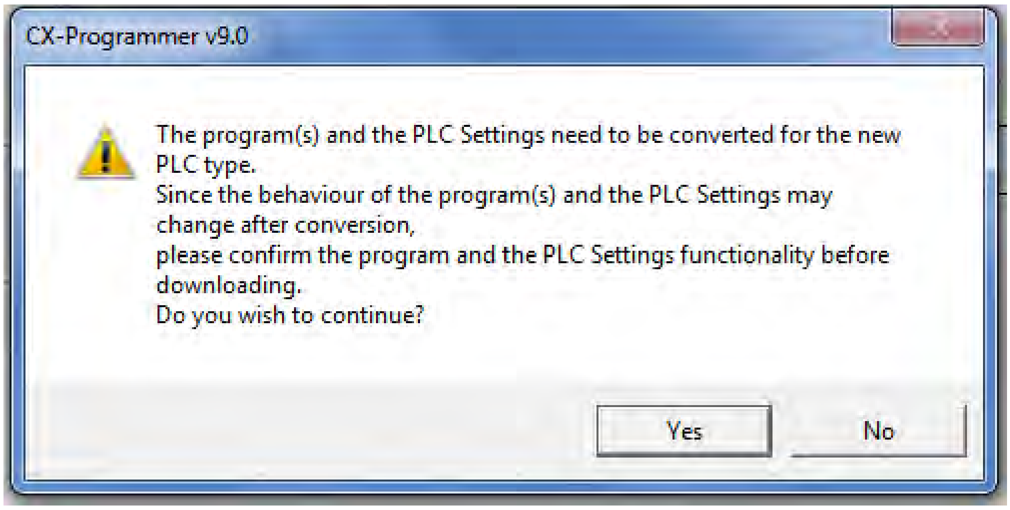


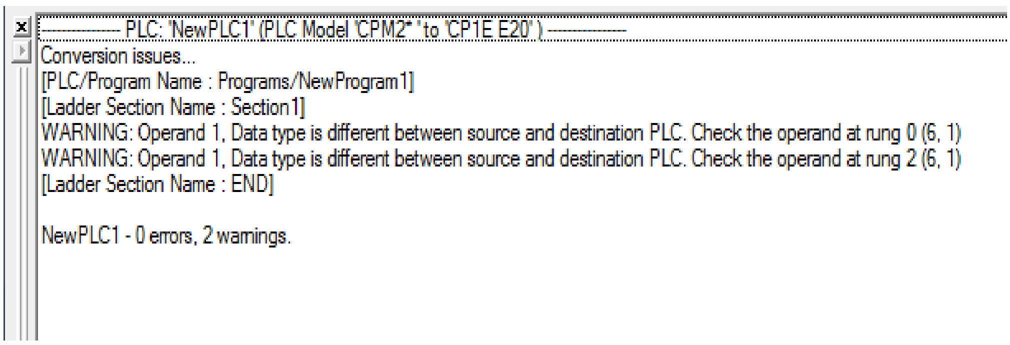
Setelah diklik OK akan muncul kembalik tampilan seperti gambar x, dan klik OK

Sekarang program siap dihubungkan dengan PLC dan ditransfer untuk kemudian dijalankan.


Change PLC
Device Name
NewPLC1
Device Type
CPM2*
Settings...
CPM2*
CPM2*-S*
CQM1
CQM1H
CS1D-H
CS1D-S
CS1G/CJ1G
CS1G-H
Settings...
OK
Cancel
Help


Change PLC
Device Name
NewPLC1
Device Type
CPM2*
Settings...
CJ1H-H
CJ1M
CJ2H
CP1E
CP1H
CP1L
CPM1(CPM1A)
CPM2*
Settings...
OK
Cancel
Help







C. Lembar Kerja

APLIKASI PLC PADA RANGKAIAN LISTRIK

a) PENDAHULUAN

Perkembangan industri dewasa ini, khususnya dunia industri di negara kita berjalan amat pesat seiring dengan meluasnya jenis produk-produk industri, mulai dari apa yang digolongkan sebagai industri hulu sampai dengan industri hilir. Kompleksitas pengolahan bahan mentah menjadi bahan jadi yang berproses baik secara fisika maupun secara kimia, telah memacu manusia untuk selalu meningkatkan dan memperbaiki untuk kerja sistem yang mendukung proses tersebut, agar semakin produktif dan efisien. Salah satu yang menjadi perhatian utama hal ini adalah penggunaan sistem pengendalian proses industri.

Dalam era industri modern, sistem kontrol, proses industri biasanya merujuk pada otomatisasi sistem kontrol yang digunakan. Sistem kontrol industri di mana peranan manusia masih amat dominan (misalnya dalam merespons besaranbesaran proses yang diukur oleh sistem kontrol tersebut dengan serangkaian langkah berupa pengaturan panel dan saklar-saklar yang relevan) telah banyak digeser dan digantikan oleh sistem kontrol otomatis. Sebabnya jelas mengacu pada faktor-faktor yang mempengaruhi efisiensi dan produktivitas industri itu sendiri, misalnya faktor human error dan tingkat keunggulan yang ditawarkan sistem control tersebut.

Salah satu sistem kontrol yang sangat luas penggunaannya ialah Programmable Logic Controller (PLC). Penerapannya meliputi berbagai jenis industri mulai dari industri rokok, otomotif, kertas bahkan sampai pada industri tambang, misalnya pada pengendalian turbin gas dan unit industri lanjutan hasil pertambangan. Kemudahan transisi dan sistem kontrol sebelumnya (misalnya dari sistem kontrol berbasis relay mekanis) dan kemudahan trouble-shooting dalam

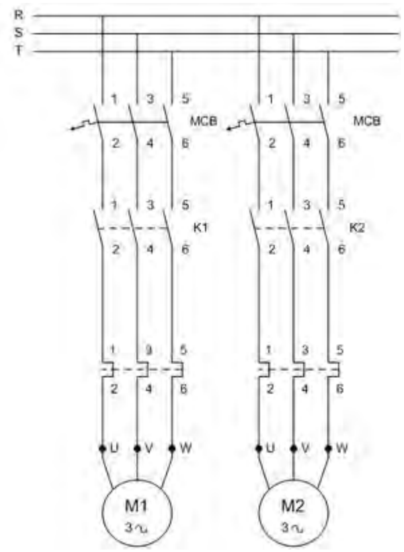
konfigurasi sistem merupakan dau faktor utama yang mendorong populernya PLC.

b) ALAT DAN BAHAN

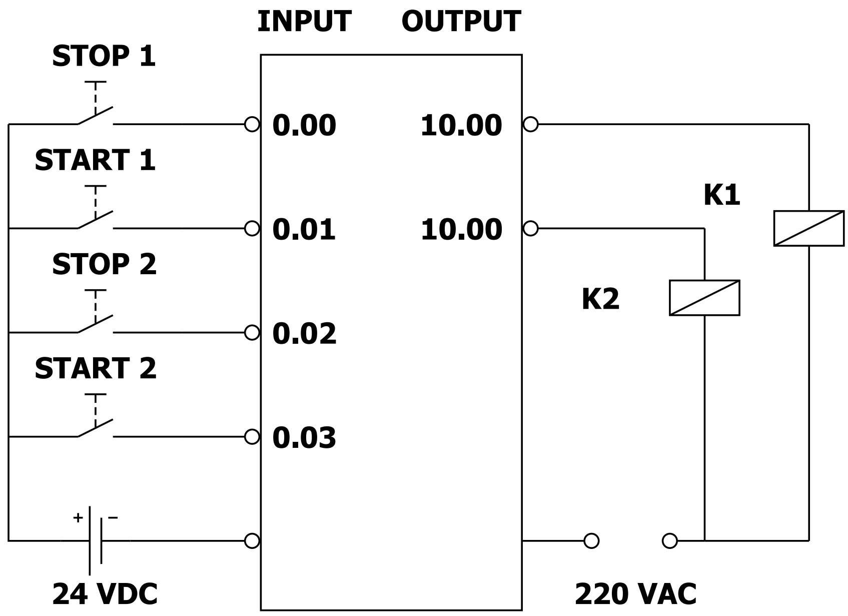
1) Trainer PLC
2) Trainer pengendali Magnetik
3) PC Komputer / Laptop
4) Kabel Jamper
5) Obeng

c) LANGKAH KERJA DAN TUGAS

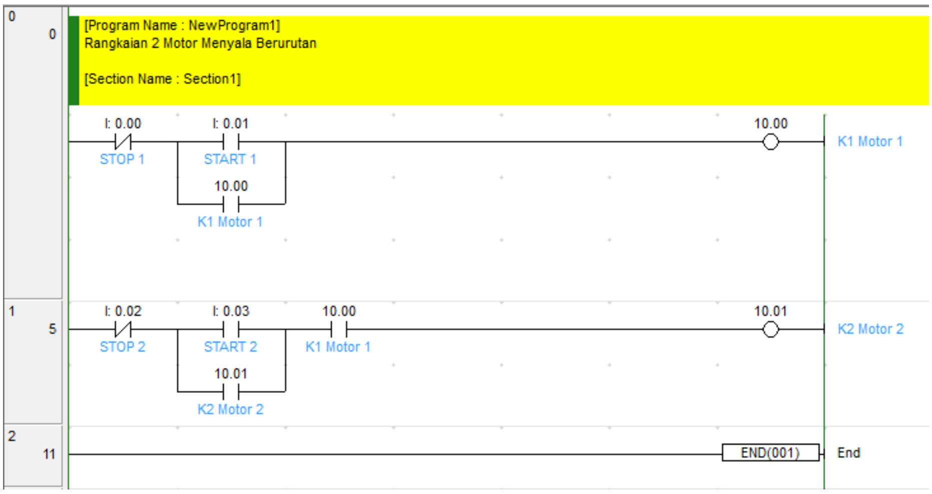
1) Buatlah aplikasi PLC untuk Rangkaian 2 Motor Kerja Berurutan seperti ditunjukan gambar dibawah !



Gambar 88 Diagram Daya 2 Motor 3 Phasa Menyala Berurutan

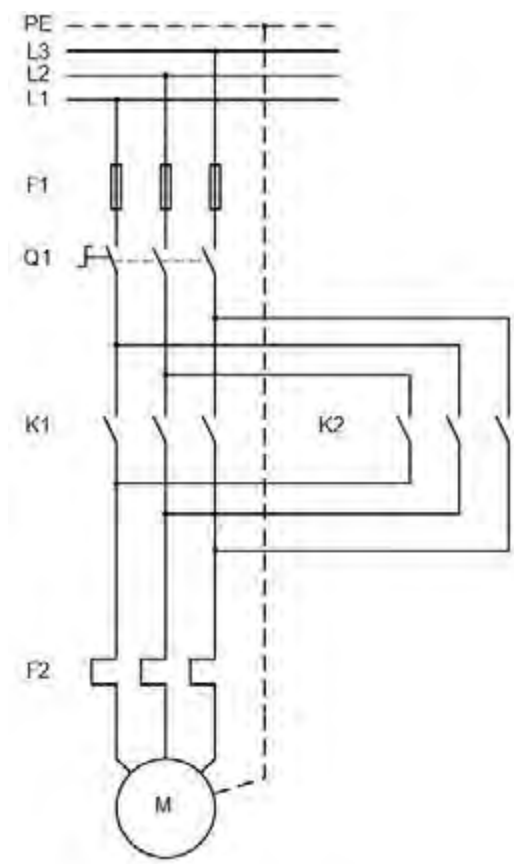


Gambar 89 Wiring PLC

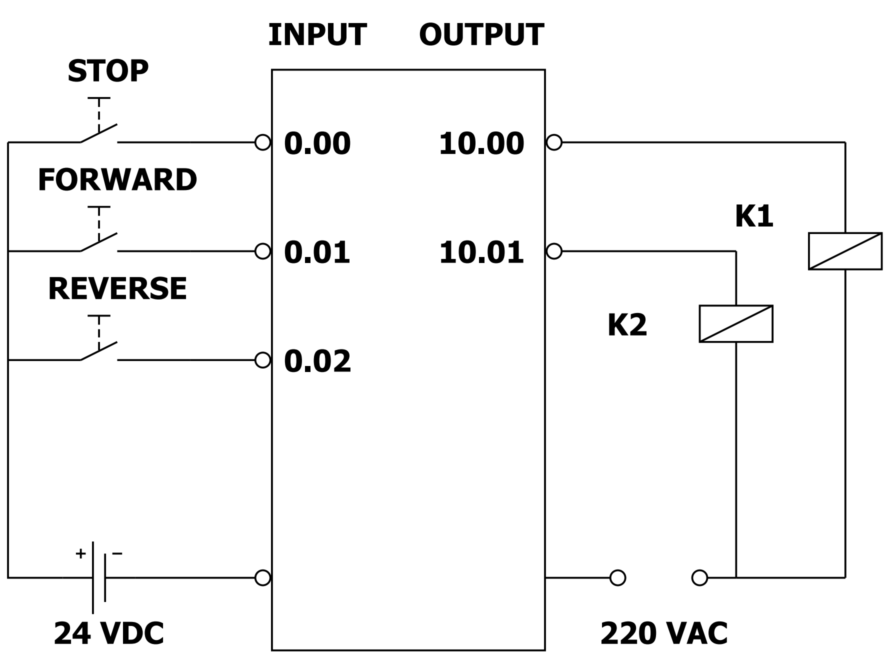


Gambar 90 Program *Ladder* Diagram PLC

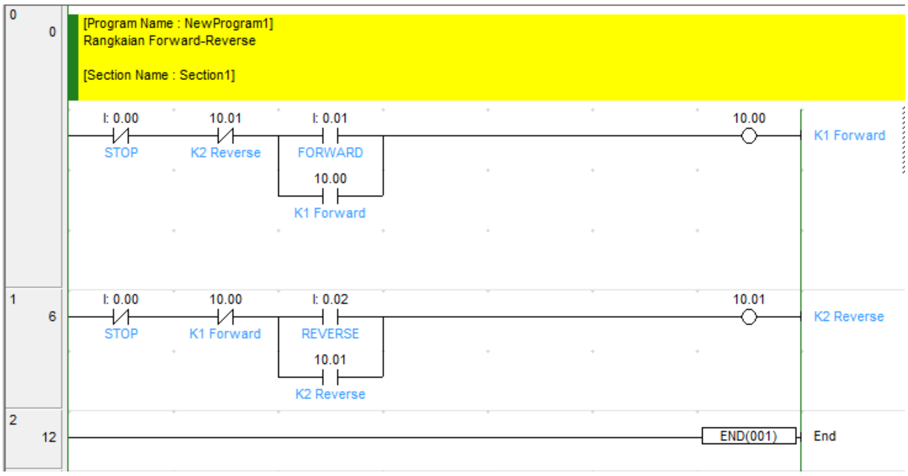
2) Buatlah aplikasi PLC untuk Rangkaian Motor 2 Arah Putaran (Forward – Reverse)



Gambar 91 Diagram Daya Forward-Reverse

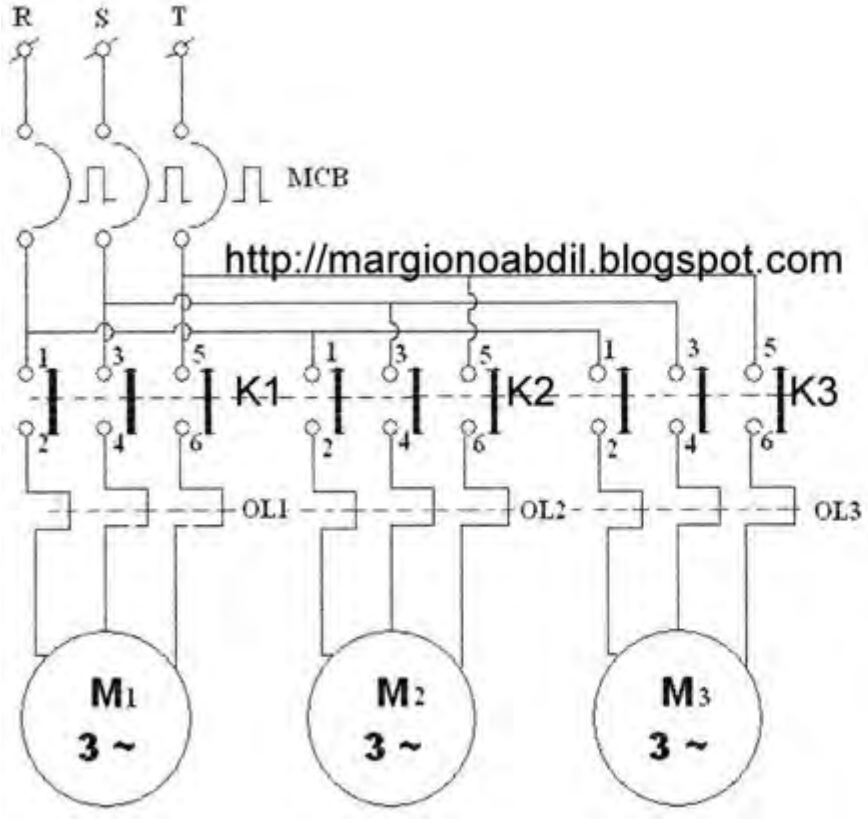


Gambar 92 Wiring PLC

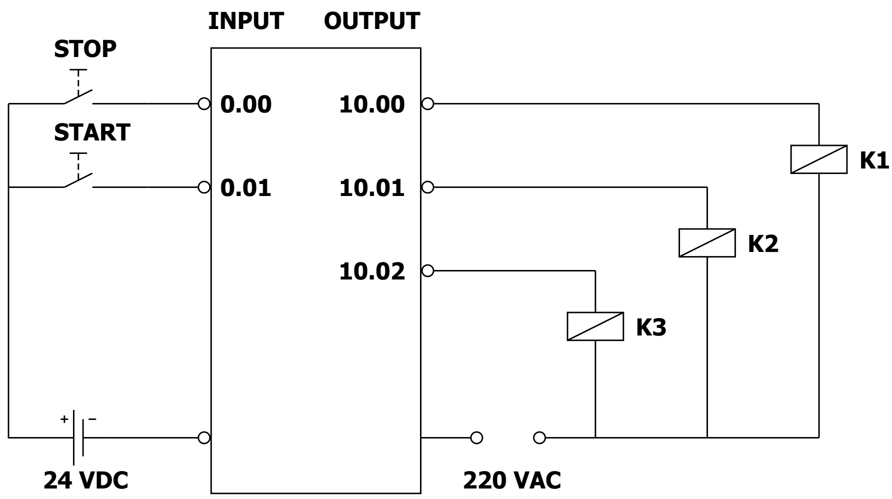


Gambar 93 Program *Ladder* Diagram PLC

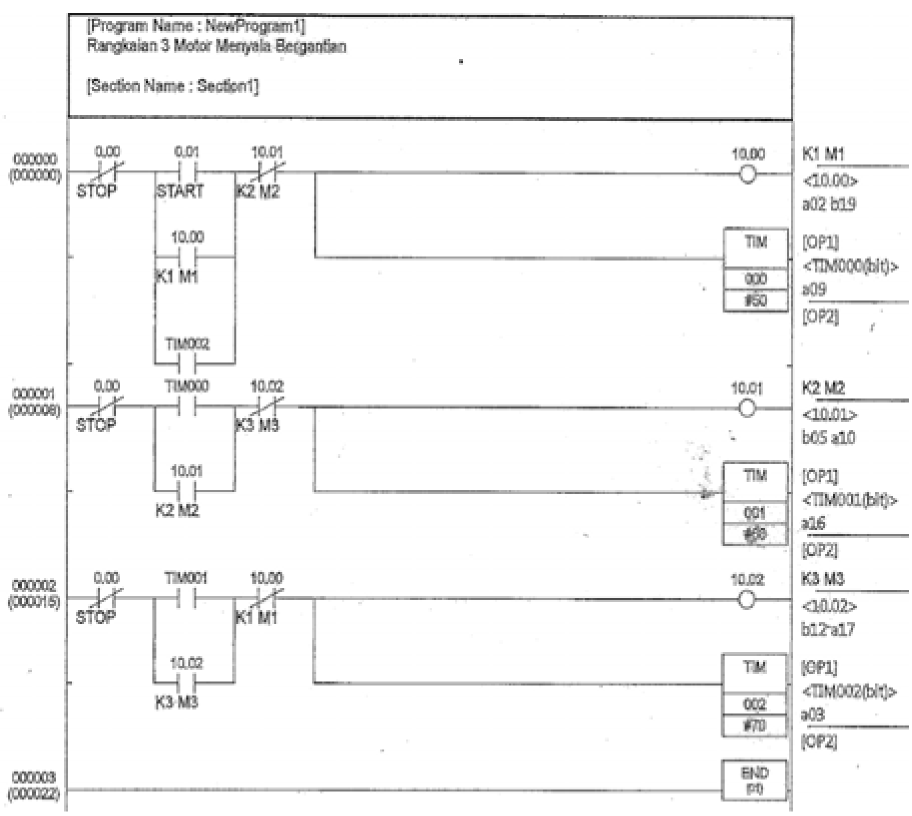
3) Buatlah aplikasi PLC untuk Rangkaian 3 Motor Kerja Bergantian



Gambar 94 Diagram Daya 3 Motor Bekerja Bergantian

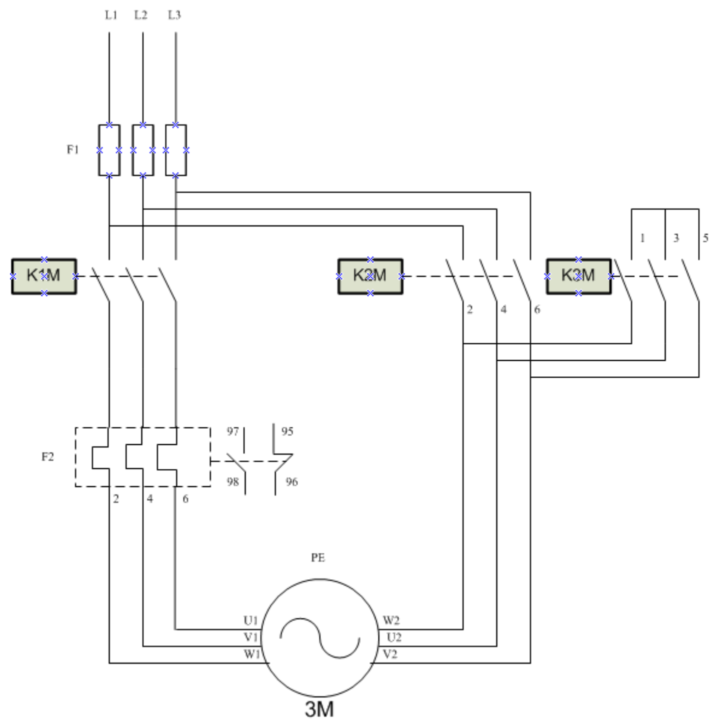


Gambar 95 Wiring PLC

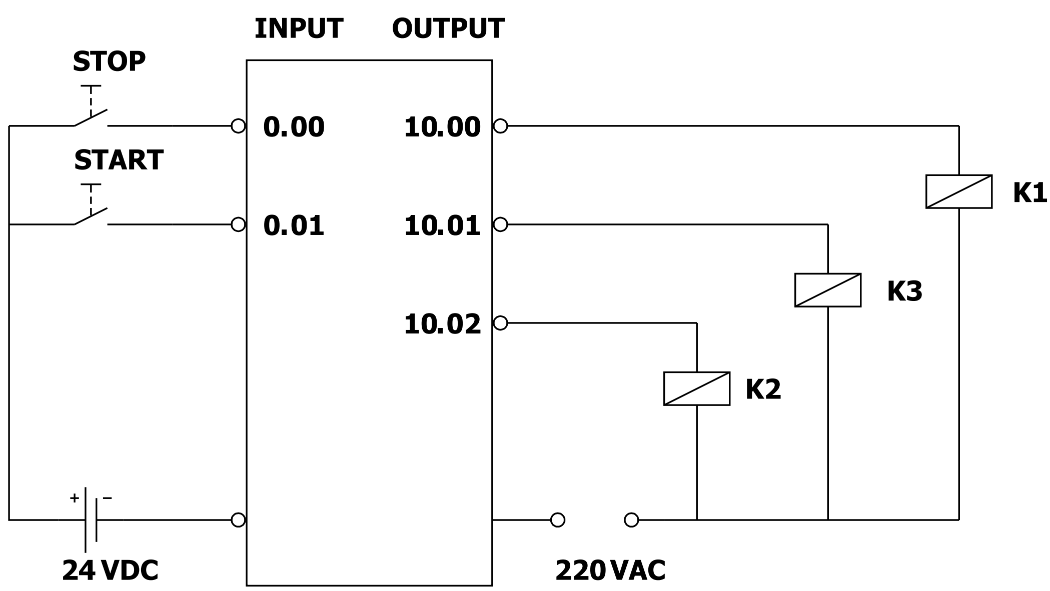


Gambar 96 Program *Ladder* Diagram PLC

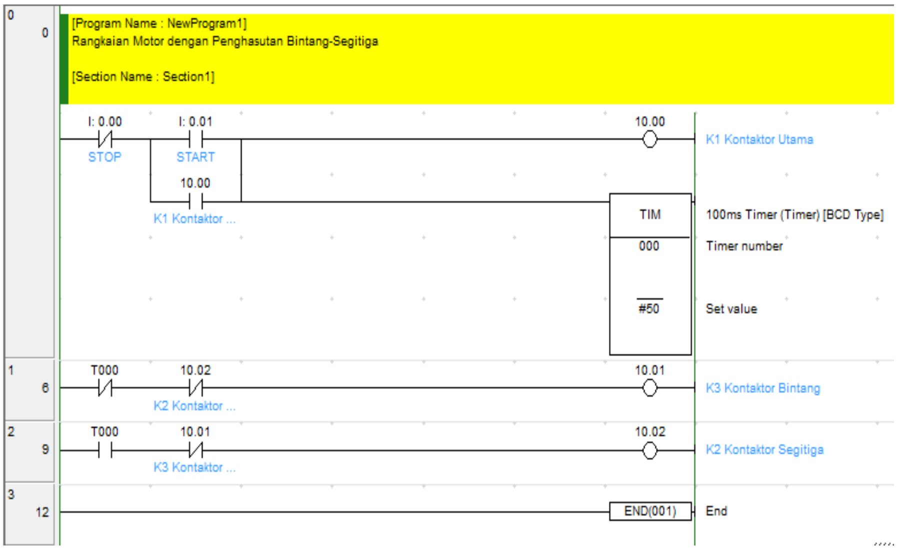
4) Buatlah aplikasi PLC untuk Rangkaian Motor dengan Penghasutan Bintang – Segitga (Y – Δ)



Gambar 97 Diagram Daya Motor dengan Penghasutan Bintang – Segitga (Y – Δ)



Gambar 98 Wiring PLC



Gambar 99 Program *Ladder* Diagram PLC

d) KESELAMATAN DAN KESELAMATAN KERJA

1) Gunakanlah pakaian praktik.
2) Bacalah dengan seksama dan benar petunjuk praktikum.
3) Hati-hati dengan aliran arus listrik.
4) Jangan meletakkan peralatan di tepi meja.
5) Kabel penghubung yang tidak terpakai jangan dekat dengan rangkaian.
6) Tanyakan kepada instruktur hal-hal yang meragukan.

DAFTAR PUSTAKA

1. M. Budiyanto, A. Wijaya, 2003, **Pengenalan Dasar-Dasar PLC,** Gava Media,
2. Yogyakarta.
3. Indonesia Australia Partnership for Skills Development
4. Batam Institutional Development Project.
5. **Chandra MDE,** http://telinks.wordpress.com/2008
6. Blogger, Drs.Agus Subowo K3TITL UPTD Pendidikan SMK Negeri 1 Bangil.
7. Gultom Anton,Modul Otomasi Industri Pemrograman PLC,2011.
8. Modul PLC Production System and Automation Laboratory-STT Telkom Bandung
9. Modul PLC Basic Training Manual OMRON,2002,
10. ____ ,1996,**SYSMAC CQM1/CPM1 Programmable Controller Programming**
11. **Manual,** OMRON Asia Pacific, PTE, Ltd, Singapore.
12. ____ ,1997,**CPM1A, Programmable Controllers Operation Manual,** Omron
13. Corporation Systems Components Division, Tokyo.
14. ____ ,1997,**CPM2A, Programmable Controllers Operation Manual,** Omron
15. Corporation FA Systems Division, Shizuoka.
16. ____ , 1999, **Beginner's Guide to PLC,** OMRON Asia Pacific, PTE, Ltd, Singapore.
17. ____ , 2001, **CX-Programmer User Manual Version 2.1**
18. ____ , 2001, **CX Server Run Time Version 1.6**